NASA CONTRACTOR REPORT 177478

AUTONOMOUS FLIGHT AND REMOTE SITE LANDING GUIDANCE RESEARCH FOR HELICOPTERS

R. V. DENTON
N. J. PECKLESMA
F. W. SMITH

TAU CORPORATION
485 ALBERTO WAY, BUILDING D
LOS GATOS, CA 95030

PREPARED FOR
AMES RESEARCH CENTER
UNDER CONTRACT NAS2-12180

NASA
National Aeronautics and
Space Administration

NASA CONTRACTOR REPORT 177478

AUTONOMOUS FLIGHT AND REMOTE SITE LANDING GUIDANCE RESEARCH FOR HELICOPTERS

R. V. DENTON
N. J. PECKLESMA
F. W. SMITH

CONTRACT NAS2- 12180
AUGUST 1987

AUTONOMOUS FLIGHT AND REMOTE SITE LANDING GUIDANCE RESEARCH FOR HELICOPTERS

Contract NAS 2-12180

FINAL REPORT

AUGUST 1987

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (continued)

ABBREVIATIONS

AGC	Automatic Gain Control
AGL	Above Ground Level
ALV	Autonomous Land Vehicle
CADC	Central Air Data Computer
CSP	Common Signal Processor
DLMS	Digital Land Mass
DP	Dynamic Programming
FLIR	Forward-Looking Infrared
GN&C	Guidance Navigation Control
GPS	Global Positioning System
HGC	Horizontal Command Generator
HUD	Head Up Display
INS	Inertial Navigation System
IR&D	Independent Research & Development
LHX	Light Helicopter Experimental
LLTV	Low Light Television
NOE	Nap-of-the-Earth
OA	Obstacle Avoidance
SAR	Synthetic Aperture Radar
SBIR	Small Business Innovative Research
SOW	Statement of Work
TF/TA	Terrain Following/Terrain Avoidance
VGC	Vertical Command Generator
VLSI	Very Large Scale Integrated Circuits
VT	Transit distance between image frames

1.0 SUMMARY

1.1 INTRODUCTION

This Final Phase I Technical Report documents the results of work performed for the NASA-Ames Research Center under NASA Contract NAS2-12180, Autonomous Flight and Remote Site Landing Guidance Research for Helicopters. This research has focused on the automation of the Guidance, Navigation and Control (GN&C) functions for low altitude flight and remote site landings.

Autoguidance technology is an important research area with several applications. For NASA, it is an element of the overall aircraft automation program which has been pursued for several years. This technology points to a reduction in pilot workload in both civilian and military operations. Low altitude flight, particularly at NOE elevations, can be a high stress operating environment in which reaction times are minimal and tolerance for error nonexistent. The advancement of this technology may reduce the potential for accidents through automated obstacle detection and avoidance. This type of capability may be critically essential for military single pilot operations such as in LHX.

1.2 RESEARCH OBJECTIVE

The objective of this study was to conduct research that has the potential of leading to automated low-altitude flight and landing in remote areas within a civilian environment, where initial cost, on-going maintenance costs, and system productivity are important considerations. An approach has been implemented which has: 1) utilized those technologies developed for military applications which are directly transferable to a civilian mission; 2) exploited and developed technology areas where new methods or concepts are required; and 3) undertaken research identified as having the potential of leading to innovative methods or concepts required to achieve a manual and fully automatic remote area low-altitude and landing capability. The project has resulted in a definition of a system operational concept that includes a sensor subsystem, a sensor fusion/feature extraction capability, and a guidance and control law concept. These subsystem concepts have been developed to sufficient depth to enable further exploration within the NASA simulation environment, and to support programs leading to flight test.

1.3 SCOPE/GUIDELINES/REQUIREMENTS

Part of the tasks in the Phase I contract were accomplished by directly applying the results of similar military sponsored programs. Table 1 highlights the related technologies of two programs which closely interface with this effort. The Autonomous Land Vehicle program accommodates sensors and sensor application techniques which are very similar to those needed for obstacle avoidance in an NOE operational environment. The Pilot's Associate Program addresses the automation of aircraft trajectory and the ways of best supplying this information to the pilot.

The majority of the work conducted under this contract should be considered innovative research in task areas that called for additional effort. It was done from a broad systems requirements and definition point of view, where tradeoffs and in-depth analysis in immature technology areas could be reviewed before proceeding to simulation.

This report defines the concepts that were developed and summarizes the effectiveness of the concepts based on computer analysis. It is expected that these concepts might be further evaluated by NASA in piloted simulations, and potentially in flight test.

Project Approach Overview

For this project, the research has been directed towards advancing rotorcraft performance and effectiveness, particularly in guidance and control functions for NOE flight and remote site landing in conditions which may include poor visibility and darkness. The project approach emphasizes the following: (1) identify guidance approach; (2) identify sensor imaging and machine vision techniques; (3) push promising technologies.

The initial part of this study identifies the system concept which relates navigation, sensors, and guidance and control. In the area of integrated navigation, a design was put together for a helicopter landing mission-tailored landing algorithm. The developed navigation sensor blending algorithm emphasizes Global Positioning System (GPS) and Inertial Navigation System (INS) for data sources, considering radar altimeter inputs. Concepts were then explored with regard to the sensor interface with helicopter landing guidance algorithms. Sensing requirements were then developed for primary and secondary rotorcraft missions. A preliminary system architecture was developed which would allow for the synthesis, integration, and augmentation of the involved technologies. An approach was then designed to enable the blending of sensor data such that the essential guidance information is extracted for both manual and fully automatic flight. Guidance and control laws were subsequently developed using the information derived from the sensor data. The product of these algorithms provide trajectory information capable of being used by an autopilot or displayed on a HUD in a manner that could be effectively used by the pilot for automated terrain flight. The work completed under each of the tasks was evaluated for proposed concept effectiveness.

ORIGINAL PAGE IS
OF POOR QUALITY

AUTONOMOUS LAND VEHICLE	PILOT'S ASSOCIATE
SENSORS FLIR TV LASER RADAR ACOUSTIC	NOT ADDRESSED IN DETAIL
SENSOR PROCESSING AND BLENDING LIMITED DOMAIN INITIALLY (ROAD FOLLOWING) LANDMARK RECOGNITION (VISION) FOR NAVIGATION UPDATING LATER IN PROGRAM	ADDRESSED AT AN "INFORMATION CONTENT" LEVEL ONLY
EXPERT SYSTEMS SYSTEM EXECUTIVE AND "MAP-MAKING" MAY BE USEFUL	HEAVY EMPHASIS ON SYSTEM EXECUTIVE, WHICH THEN CALLS SUBSYSTEM "EXPERTS." MULTIPLE COOPERATIVE EXPERT SYSTEMS.
PLANNING BOTH FAR FIELD AND NEAR FIELD PLANNING NEAR FIELD MAY NEED MORE "INTELLIGENCE" THAN IN NOE APPLICATION OBSTACLE AVOIDANCE IN VERY NEAR FIELD	INTENT WAS TO INCLUDE AUTOMATIC MISSION PLANNING FUNCTIONS. NOT KNOWN IF "NEAR FIELD" IS BEING TREATED.
PILOT VEHICLE INTERFACE	COMPUTER-GENERATED VOICE, NATURAL LANGUAGE INTERFACE, SPEECH RECOGNITION, ADVANCED DISPLAYS.

Table 1. Relationship of NOE Research to DARPA Strategic Computing Program Application

Project Goals

The objectives in conduct of this research have been (1) to provide NASA access to current technologies; (2) to synthesize, integrate and augment these technologies and define a feasible, automated GN&C system; and (3) to demonstrate the effectiveness of this system through analysis, simulation and possible flight test. The resultant GN&C system has incorporated the best available technologies and has provided the basis for a practical, implementable and affordable mechanization.

NASA requested that the research effort focus on a relatively small set of lower cost sensors when addressing Tasks II (Sensor Requirements) and Task III (Blending of Sensor Data). It is understood that this requirement is driven by the procurement aim of developing related technology that has the potential of being used in the civil environment where considerations of cost may be significantly important.

1.4 PREVIEW OF RESULTS

The results of this work are significant in that a viable method has been identified and evaluated which uses relatively low cost passive sensors and digital maps to support safe helicopter flight in the Nap-of-the-Earth (NOE) flight mode.

Briefly stated, the products of the autoguidance research effort have been primarily ones of: (1) furthering the understanding of how to use passive sensors in the NOE flight environment and; (2) advancing the understanding of key guidance and control implications associated with autonomous control of a helicopter.

A method has been identified where a single video sensor (FLIR, LLTV) can be employed performing passive ranging and obstacle avoidance in a difficult real-time computing environment using image processing techniques. A promising method has been identified and specifications have been generated as to the types of hardware components and algorithms which are needed for the next stage of study. In addition, the major obstacles to further research have been identified and the systems error contributions have been quantified.

The guidance and control research efforts under this contract have resulted in the delivery of real-time NOE trajectory generating code which has since been used in man-in-loop simulations in the NASA Ames facility. Other research has coordinated the interactions of airframe and sensors to quantify the general sensor requirements, and the interrelationship, at the system level, of the sensors, sensor algorithms, navigation, and guidance of the helicopter.

2.0 RESEARCH AND RESULTS

2.1 TASK I: SYSTEM CONCEPT

The objective of this task was to define in general terms, a total system concept that will allow a helicopter to operate at low-altitude and land at unprepared sites in remote areas at night and in poor visibility conditions. The concept includes a sensor subsystem, a sensor fusion/feature extraction capability, and a guidance and control law concept. The derived concept accommodates diverse sensors and supports multiple flight modes.

2.1.1 Design Approach

The design approach to the development of the overall system concept is described as follows:

a. Formulate an initial design concept, including system operational modes and top-level block diagrams which encompass these modes, very early in the program. This early conceptualization of the system serves as a reference framework within which the workability of various subsystem designs can be evaluated.

b. During formulation of the initial system concept, draw heavily from recent and ongoing programs in the military environment to establish key subsystems and their relationships. In particular, review such military programs as Pave Pillar and TF/TA programs.

c. Revise the system concept as necessary after individual design trade studies are performed in Tasks II through IV. It is primarily at the subsystem level that alternative approaches are available to accomplish the various subsystem functions, and key trades will be accomplished during the project to identify particular approaches that appear most viable and/or which constitute lower risk in meeting overall system objectives.

2.1.2 Objectives

The objectives associated with Task I are:

a. Determine the available DoD technologies which can be exploited.

b. Develop a system concept at the subsystem block diagram level.

c. Establish the framework for a detailed design.

2.1.3 Research Plan

The initial plan for determining the system concept occurred in two phases. The first phase culminates in the development of an initial system concept and the identification of the set of operational modes for the system. The steps that were be followed to accomplish this are:

- Identify existing and planned DoD programs and published research that are usable in the current context.
- Identify alternative system concepts.
- Adapt the alternative system concepts to existing technologies.
- Select an initial system concept and a set of operational modes.
- Document the selected system concept with a block diagram showing subsystem control and interactions.

This system block diagram served as a strawman framework to which Tasks II, III and IV were worked. While Tasks II, III and IV were being worked, the system concept was allowed to evolve to accept new technologies and to reflect the maturation of other technologies. This second, evolutionary phase culminated in a final system concept that accurately reflects the results developed by Tasks II, III and IV. Figure 1 reflects that plan.

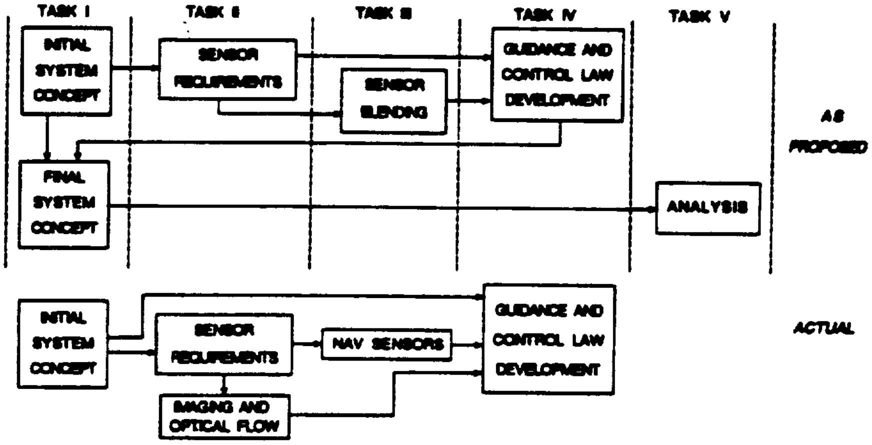


Figure 1. Phase I Task Flows in Project Work Plan

The initially proposed Task V analysis functions were incorporated into Tasks II & IV. For example, as the imaging and optical flow concepts were developed, image data was obtained and used to further the understanding of the processing techniques separately from the overall system concept. Likewise, the guidance and control law development was performed with a recognition of the interface requirements with respect to the sensors, but without assembling an integrated end-to-end simulation. As a result, the system concept was analyzed to a degree of detail which was sufficient to verify it's viability.

As stated earlier, an important aspect of the approach to Task I is to formulate a system concept very early in the program, and to perform subsystem design and analysis tasks in the context of this system concept, so as to evolve it into a complete, coherent and implementable system. Formulation of the system concept requires a definition of flight phases and nodes of operation, and a definition of system requirements for each phase/mode. These system requirements include:

- functions to be accomplished,
- associated timing and accuracy requirements,
- attendant requirements on subsystems such as sensors, and
- requirements for crew interface.

2.1.4 System Operational Modes

A starting point for development of the preliminary system concept is to identify the mission functions to be accomplished. The system operational modes are typically thought of as a combination of flight phase or regime, mission function, and crew interface mode (automatic or manual), as described in Table 2. For example, referring to the table, AUTO-TF/TA, MANUAL APPROACH, and AUTO-LANDING might all be thought of as system operational modes.

Accordingly, Table 2 shows a structure of the operational modes of the proposed system. For completeness, the table includes cruise as a flight regime (or phase) in addition to the flight regimes of greater emphasis--low altitude flight and approach/landing. The low-altitude flight regime has been broken down into three distinguishable sub-phases (or mission profiles):

- Terrain following/terrain avoidance (TF/TA), as might be used to fly search patterns for search and rescue missions;
- TF/TA with obstacle avoidance (TF/TA/OA), as might be necessary for NOE flight; and
- Hover, as might be used for navigational orientation or local area search.

The approach/landing flight regime in Table 2 has been similarly divided into the sub-phases of approach, hover, and landing. Automatic or manual operation is allowable for each sub-phase cited above.

As can also be noted in the table, hover has been included as a distinct sub-phase/mission profile. This capability of helicopters is, of course, a very important distinction from fixed wing aircraft, and will also likely exhibit sensor requirements and crew interface requirements which are different from the other operational modes.

While it may not be strictly necessary to define TF/TA and NOE flight as separate sub-phases or mission profiles, the distinction is carried along here because of the possibly different requirements for sensor data and crew interface. Accordingly, NOE flight is distinguished from TF/TA by its extreme

FLIGHT REGIME/ PHASE	FLIGHT SUB-PHASE/ MISSION PROFILE	TYPICAL SENSOR UTILIZATION	DISPLAY REQUIREMENTS	
			AUTO	MANUAL
Cruise	NAV	● INS ● Altimeter ● GPS	● Status, Nav	● Status, Nav
Low Altitude Flight	TF/TA	● Digital Map ● GPS ● INS ● FLIR	● Situation Awareness	● Task-Oriented
	TF/TA/OA (Nap-of-the-Earth)	● Digital Map ● GPS ● INS ● FLIR ● CO_2 Laser	● Situation Awareness	● Task-Oriented ● Bird's Eye
	Hover	● INS ● GPS ● CO_2 Laser	● Situation Awareness	● Task-Oriented ● Command Slew
Approach/ Landing	Approach	● INS ● FLIR ● GPS ● CO_2 Laser ● GPS Differential ● mmW Radar	● Situation Awareness	● Task-Oriented
	Hover	● INS ● FLIR ● GPS ● CO_2 Laser ● GPS Differential ● mmW Radar	● Situation Awareness	● Task-Oriented ● Command Slew
	Landing	● INS ● FLIR ● GPS ● CO_2 Laser ● GPS Differential ● mmW Radar	● Situation Awareness	● Task-Oriented ● Inertial

Table 2. System Operational Modes

proximity to the ground, where vegetation, wires and other obstacles are a major consideration and where correlation of active sensors to the digital terrain map may be more difficult because of restricted perspective view.

Figure 2, which summarizes a probability of clobber study [Ref. 9], shows justification for including an obstacle avoidance capability in the lowest altitude regimes. The figure illustrates that any low-level flight using only DMA digital terrain (curve A) map data with its inherent statistical nature ($\sigma = 26$m for this example) and the lack of information on obstacle locations presents a probability of clobber exceeding 10% over a representative flight trajectory. By adding cultural features to the terrain data base (curve B), an order of magnitude increase in clobber avoidance occurs, yet this still indicates unacceptable flight conditions due to the remaining unregistered obstacles in the data base.

With the addition of on-board sensing, the probability of clobber decreases by an order of magnitude if 90% of the obstacles are detected (curve C) and to acceptably low levels when all obstacles are detected with a sensor of 2 mil angular accuracy (curve D).

It may be concluded that flying according to the DMA data base alone would lead to unacceptably high probability of clobber, hence the functional requirement for a real-time sensor suite with a robust capability to detect obstacles.

2.1.5 Preliminary Block Diagrams

A top-level block diagram of the system concept is shown in Figure 3. As illustrated, the major conceptual subsystems of sensing, sensor binding, and guidance and control are roughly aligned with the corresponding research Tasks II, III, and IV of the study effort. The crew interface and sensor management tasks are shown to "straddle" Tasks III and IV because of the relevance of these management-related functions to the overall system.

The subsystem structure illustrated in Figure 3 is very general and applicable to system operation in any of the modes or flight regimes discussed in the previous section.

2.1.6 Develop System Design Methodology

The key to formulating the system concept resides in determining the sensor requirements to fly the missions as outlined in Section 2.1.3. Because of the extreme demands both in response and maneuverability associated with NOE flight, this flight mode is a driver to the overall system design.

The sensor requirements were obtained by determining the envelope of helicopter performance characteristics and maneuverability required in terrain and object avoidance flight (in the NOE mission in particular), and then applying the information to determine sensor ranges, field of view and regard, and other relevant factors. Ranging requirements and field of regard were established parametrically with regard to helicopter velocity, maneuverability, and system response time. The results of this performance analysis are then used to trade-off candidate sensor systems.

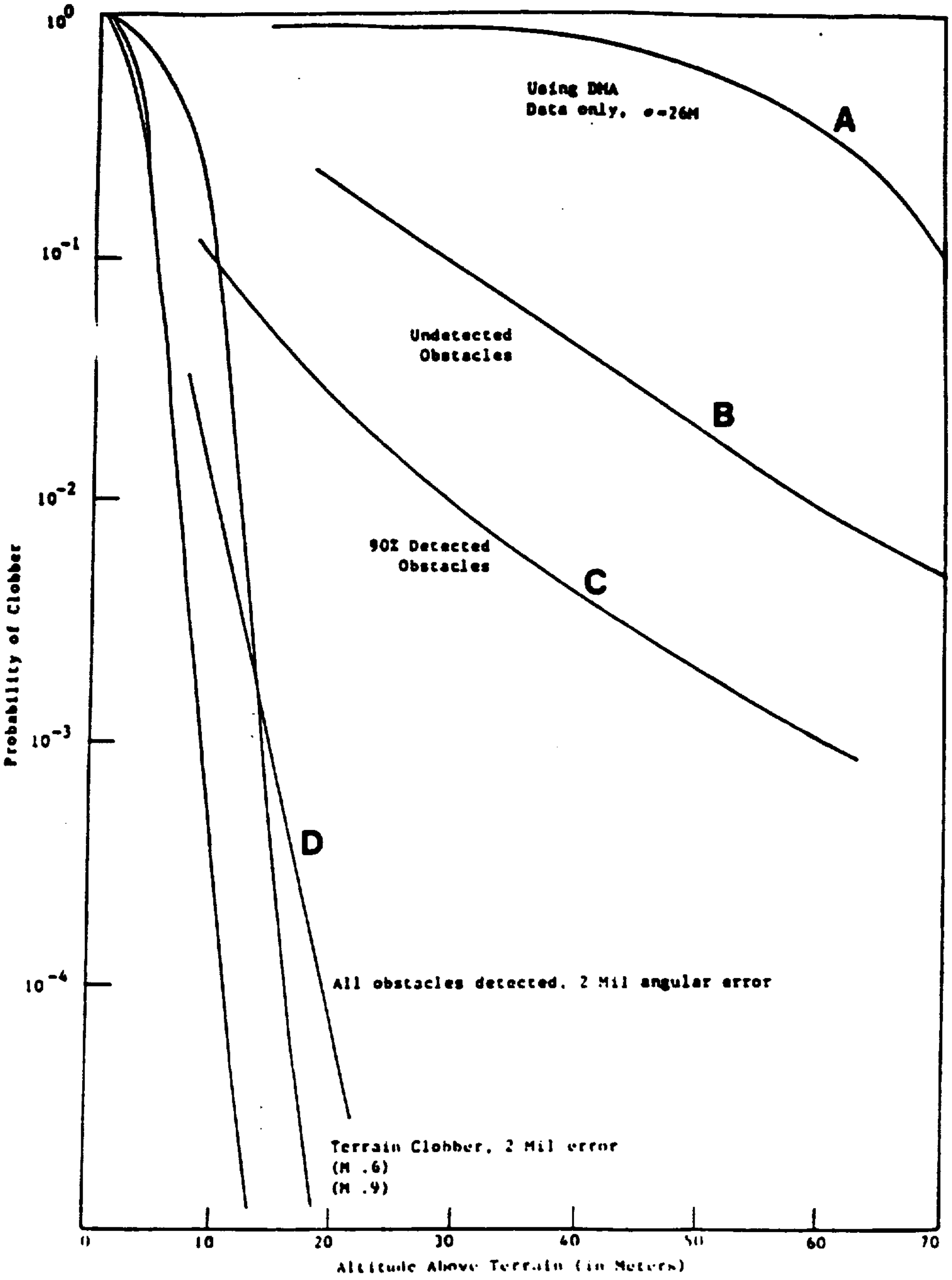


Figure 2. Probability of Clobber Versus Flight Altitude (Germany)

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

Sensor candidates were identified which are suitable to the requirements and meet with other goals. These goals include cost, availability, and a designated preference for implementing passive sensors in lieu of active sensors. In a hostile military environment, active sensors betray the aircraft position, and even in a peacetime situation, the use of lasers may require precautionary measures for the pilot and any personnel in the vicinity of the aircraft.

As the interrelationship of the sensor system was developed, a sub-system architecture was determined for the guidance portion of the concept. This architecture:

(1) Integrates the three types of navigation (very near, near and far field);

(2) Deals with selected approach to sensor fusion, namely, directed search of active sensors; and

(3) Integrates the other sensors such as GPS, INS, and digital map.

2.1.7 System Block Diagram

The final derivation of the System Block Diagram is an extension of the proposal concept. The system as shown in Figure 4 is a generic interface of Sensors, Navigation, and Guidance & Control. Each block of the diagram represents a technology, some more developed than others. This study has focused its efforts for Phase I on some of the key technologies which are just emerging in the avionics R&D environment.

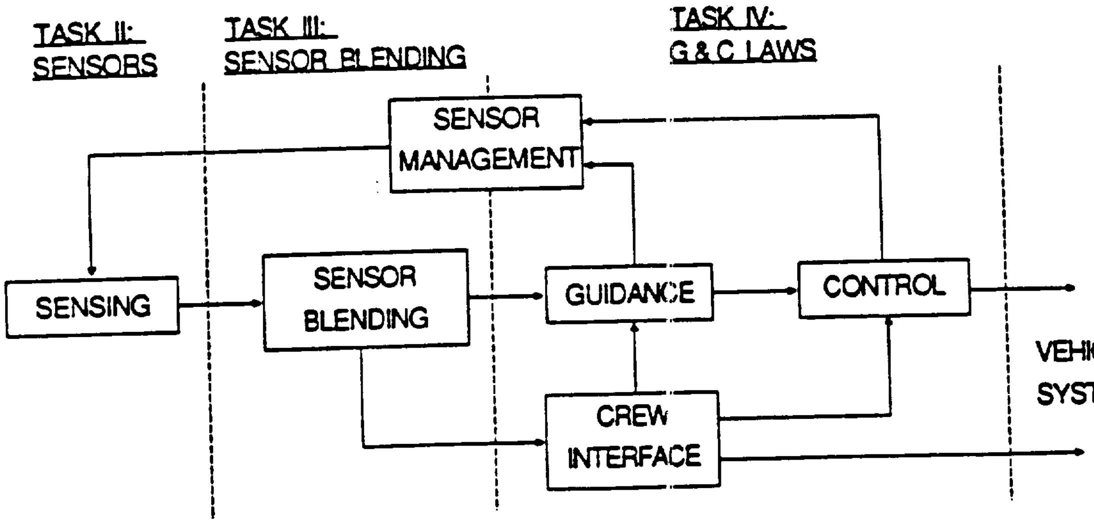


Figure 3. System Concept Block Diagram

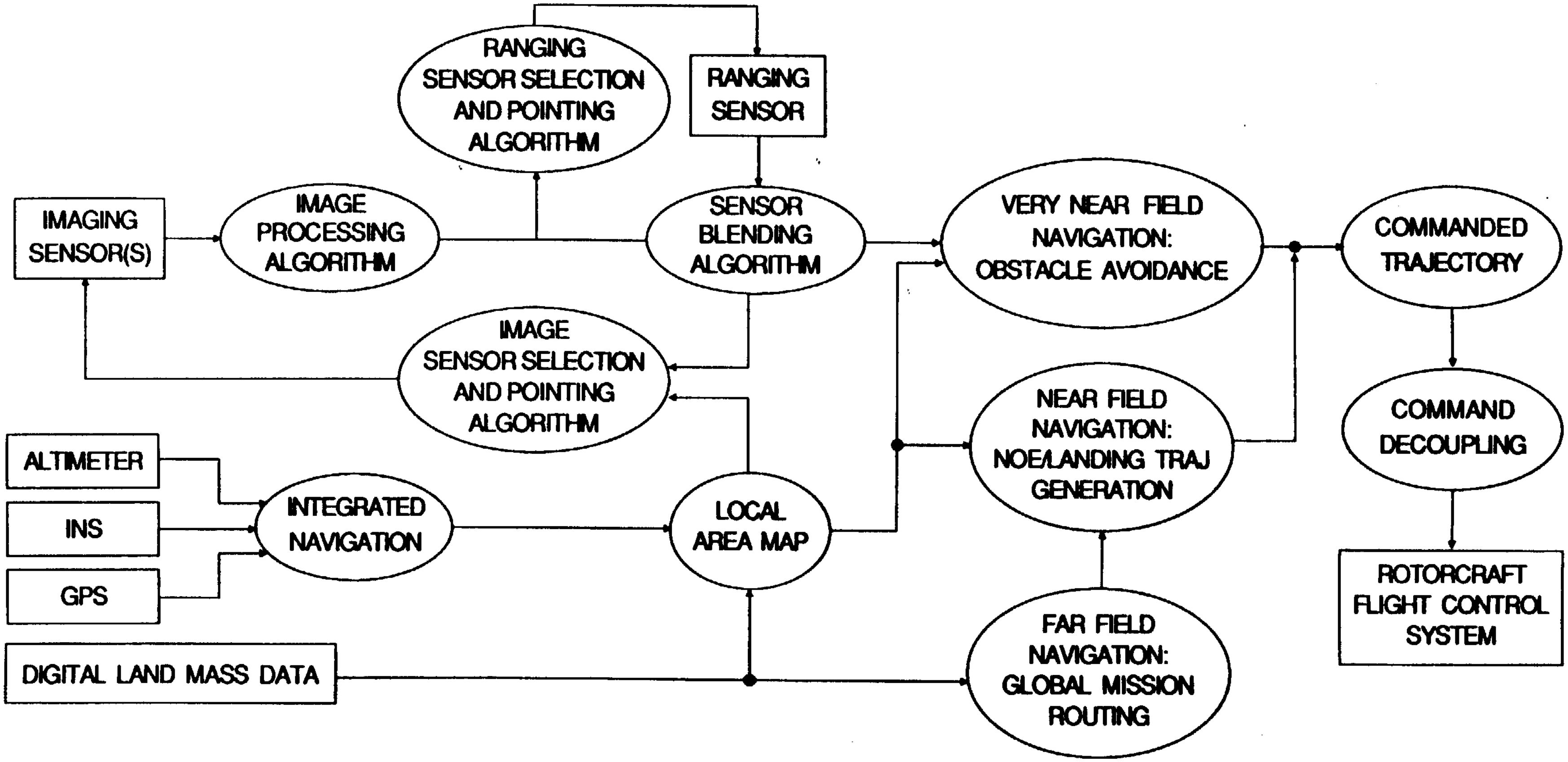

RANGING SENSOR SELECTION AND POINTING ALGORITHM
RANGING SENSOR
IMAGING SENSOR(S)
IMAGE PROCESSING ALGORITHM
SENSOR BLENDING ALGORITHM
VERY NEAR FIELD NAVIGATION: OBSTACLE AVOIDANCE
COMMANDED TRAJECTORY
IMAGE SENSOR SELECTION AND POINTING ALGORITHM
NEAR FIELD NAVIGATION: NOE/LANDING TRAJ GENERATION
COMMAND DECOUPLING
ALTIMETER
INS
GPS
INTEGRATED NAVIGATION
LOCAL AREA MAP
ROTORCRAFT FLIGHT CONTROL SYSTEM
DIGITAL LAND MASS DATA
FAR FIELD NAVIGATION: GLOBAL MISSION ROUTING

Figure 4. System Block Diagram

Integrated Navigation and Digital Maps

The blending of altimeter data, inertial navigation (INS), and GPS information into integrated navigation is an accepted technology which NASA and industry are actively working upon. Any detailed work in this area under this contract would be redundant. Digital map data is manipulated as if it were also sensor data, being fused/compared with integrated navigation to detect the possible presence of terrain obstacles and to be used in determining where to direct the onboard sensors.

Current digital map data is sparse. Commercially available DMA data sets purchased through the USGS have been found to be unreliable. However, the industry and government-wide interest in application of this technology indicates a probable availability and reliability in the future.

Sensors

It was discovered that the application of the characteristics of generic imaging and ranging sensors to the system block diagram is of more significance than the selection of individual types of sensors. The two principal sensor blending techniques which are possible are fusion and directed search. There is a basic instability in trying to fuse the azimuth and elevation information characteristic of imaging sensors with the rather broad angular measures of most ranging systems (i.e., radars).

The more flexible approach to merging the information from dissimilar sensors involves using imaging sensor data to perform separate directed search with a ranging sensor. With the appropriate sensor management algorithms, information from a suite of ranging sensors could be fused.

For example, the directed search approach allows a very narrow beam system such as a CO_2 laser to find a prioritized list of targets handed off from the imaging system. The directed search algorithm would trade-off the individual search time requirements per target with both the priority and list length to meet the scheduling requirements of its duty cycle. If the imaging system finds several targets of ranging interest, some of which are a potential clobber in the near term time span (urgent target), while others are likely to be further away and only passing near (background targets), the directed search algorithm would immediately pattern several active "pings" of the urgent target(s) and then apply the remaining time in the search cycle to the background targets.

Navigation

There are three navigation subsystems in the system concept. They are configured in close relationship where the product of one is directly applied to the next. Figure 5 illustrates the far field, near field, and very near field domains.

The far field or global mission routing subsystem has traditionally been part of mission planning prior to mission execution. Its function is to support route planning from the current location to the destination, while accounting for mission constraints such as time and fuel limitations. In recent years

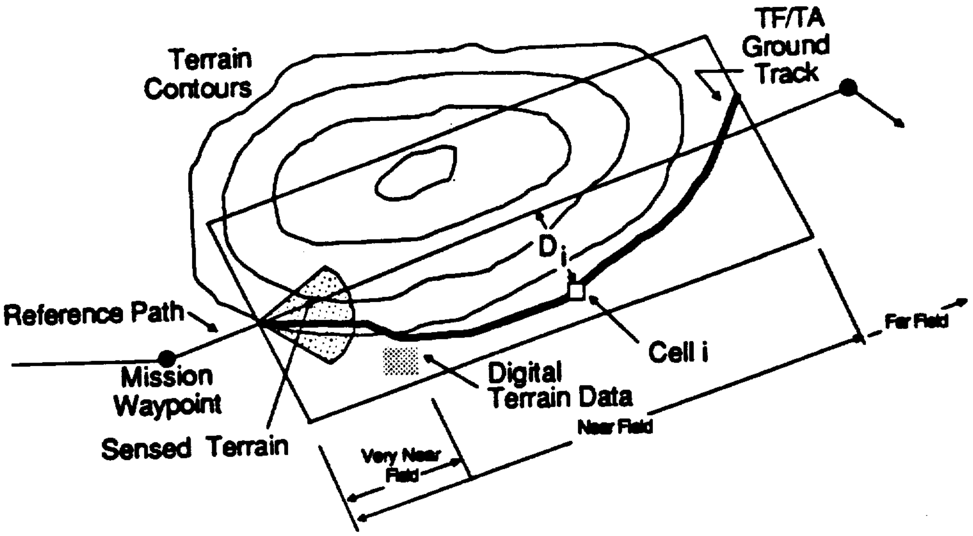


Figure 5. Navigation Concepts

there has been growing recognition that this function can be carried out automatically, during the mission, as mission requirements change.

This technology has been pursued under early internal funded research, and more recently, under another NASA SBIR Contract (NAS2-12402).

The algorithms employed to support far field global trajectory generation use a hybrid of Dynamic Programming and a recursive technique known as backward chaining. In the illustration, Figure 6, a way is indicated for enhancing this mission planning technique for real-time trajectory planning/replanning. The initial trajectory is determined using mission parameters such as known threats, waypoints, etc. and initial location and destination. A sparse grid of cost information is constructed from this data and an averaging of terrain data from the digital map.

The resultant route is then reworked on-board the aircraft using intermediate waypoint areas as destinations. A judicious selection of a subset of the map and a denser cost grid enables a refined trajectory determination over the interval.

This technique is seen as a candidate on-board far-field navigation subsystem. The near-field navigation subsystem uses both this information and available sensor data to generate a locally optimized flyable trajectory over the next 30 seconds or so of flight time. Trajectory generation applies both to the low altitude flight function as well as to the remote site landing function.

In the low altitude flight regime, this critical function is responsible for modifying the global trajectory based on the updated terrain map information to result in a new, and flyable, trajectory in the neighborhood of the global trajectory. This is portrayed in Figure 7, which shows that for the current location of the aircraft, the near-field trajectory is computed over a terrain patch immediately in front of the aircraft.

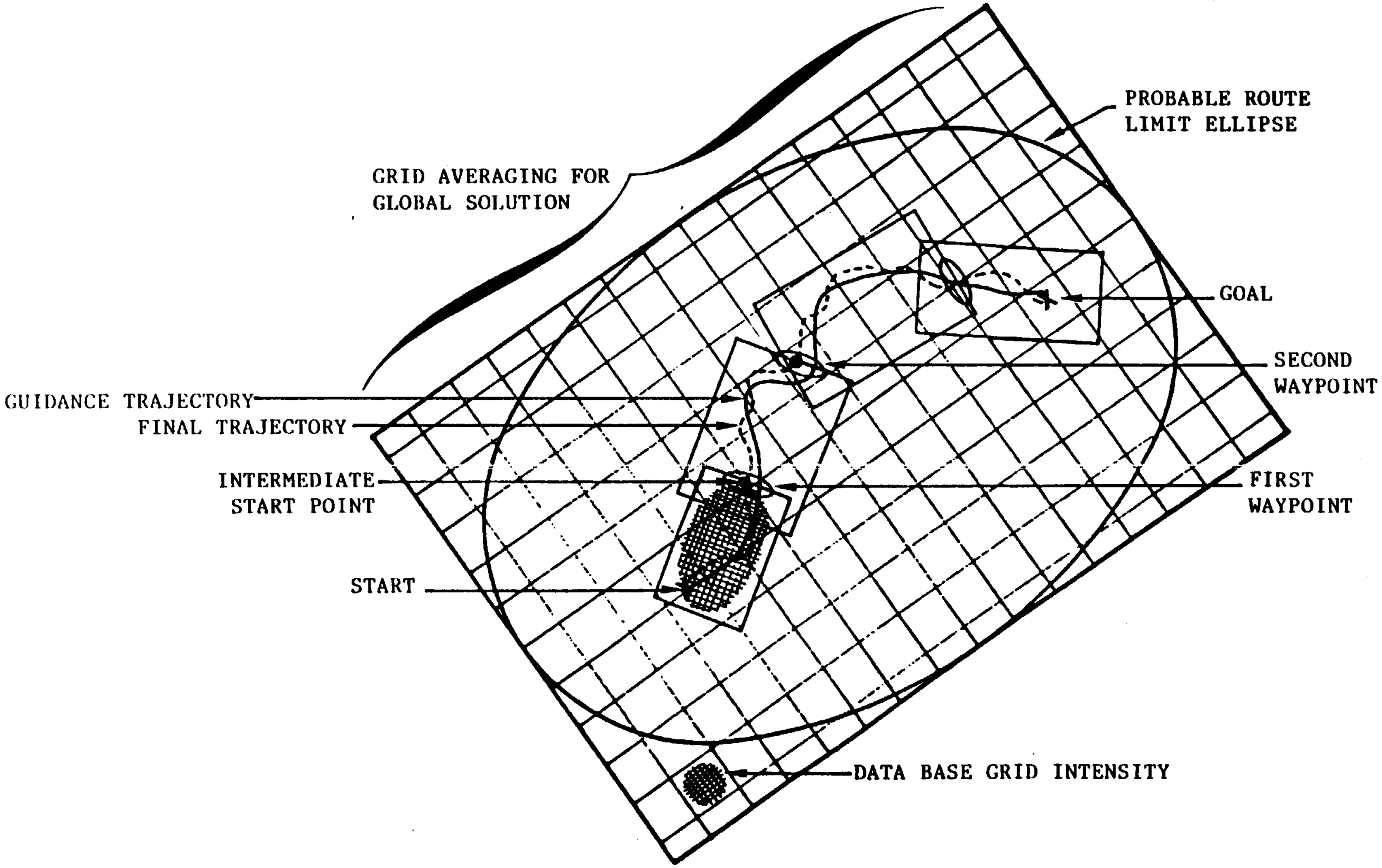
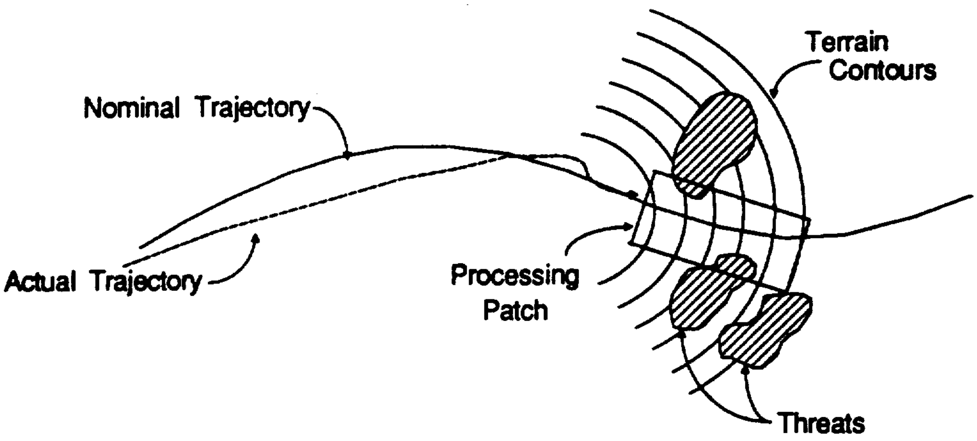


Figure 6. Far Field Navigation Concept



- Current processing path is "scrolled" along as aircraft progresses

Figure 7. Processing Path in Front of Aircraft

In this low altitude flight regime, there is an extremely small set of automatic TF/TA (and NOE) techniques that appear to be viable and which have achieved any degree of acceptance within the flight control community. One of these techniques, Dynapath, developed under an Air Force contract. Under this NASA contract, Dynapath, has been translated from this original, model coded in the C language, to FORTRAN for application at NASA. Additional modifications were made to the ways in which the Dynamic Programming controls are applied which better suit the algorithm to the relatively low speed and accelerations of helicopters versus the original application to high velocity fighter aircraft.

The details of these modifications are extensively discussed in the Task IV description.

The very near field navigation subsystem serves as an override to the nominal commanded trajectory generated by the near field navigation subsystem. When sudden obstacle detections are flagged by the sensor system and there is insufficient time to proceed with the normal computational cycle to avoid the obstacle, an override command is generated to the commanded trajectory.

2.1.8 Summary

The system block diagram developed for this phase of the contract remains in a generic state, but this is consistent with the desire to remain open to employing a relatively small set of lower cost sensors in addressing the sensor tasks, yet remaining open to future possibilities.

The rotorcraft performance envelope was defined for NOE flight. Sensing requirements were defined for the NOE flight regime which is typically at velocities of 0-60 knots and at a terrain clearance of 5-50 feet. The sensor subsystem requires a minimum field of regard of 120° and an obstacle detection capability of 420 feet, nominally, but extending over 100-1000 feet depending upon actual system response and helicopter maneuverability.

Sensor requirements for the landing mode include the same range band but require a hemispherical field of regard centered along the velocity direction. Special note should be made with regard to tail clearance in confined area landing.

For the other low altitude and approach flight modes, the sensors developed for the primary roles of NOE and landing should also serve adequately for these related modes. For contour flight, a reasonable look down capability should exist (~60°) and range measurement capability should extend to greater ranges compatible with higher expected aircraft velocities. For hovering flight, an omni-directional field of regard is highly desirable.

2.2 TASK II: SENSOR REQUIREMENTS

2.2.1 Helicopter Sensing Requirements

2.2.1.1 Introduction

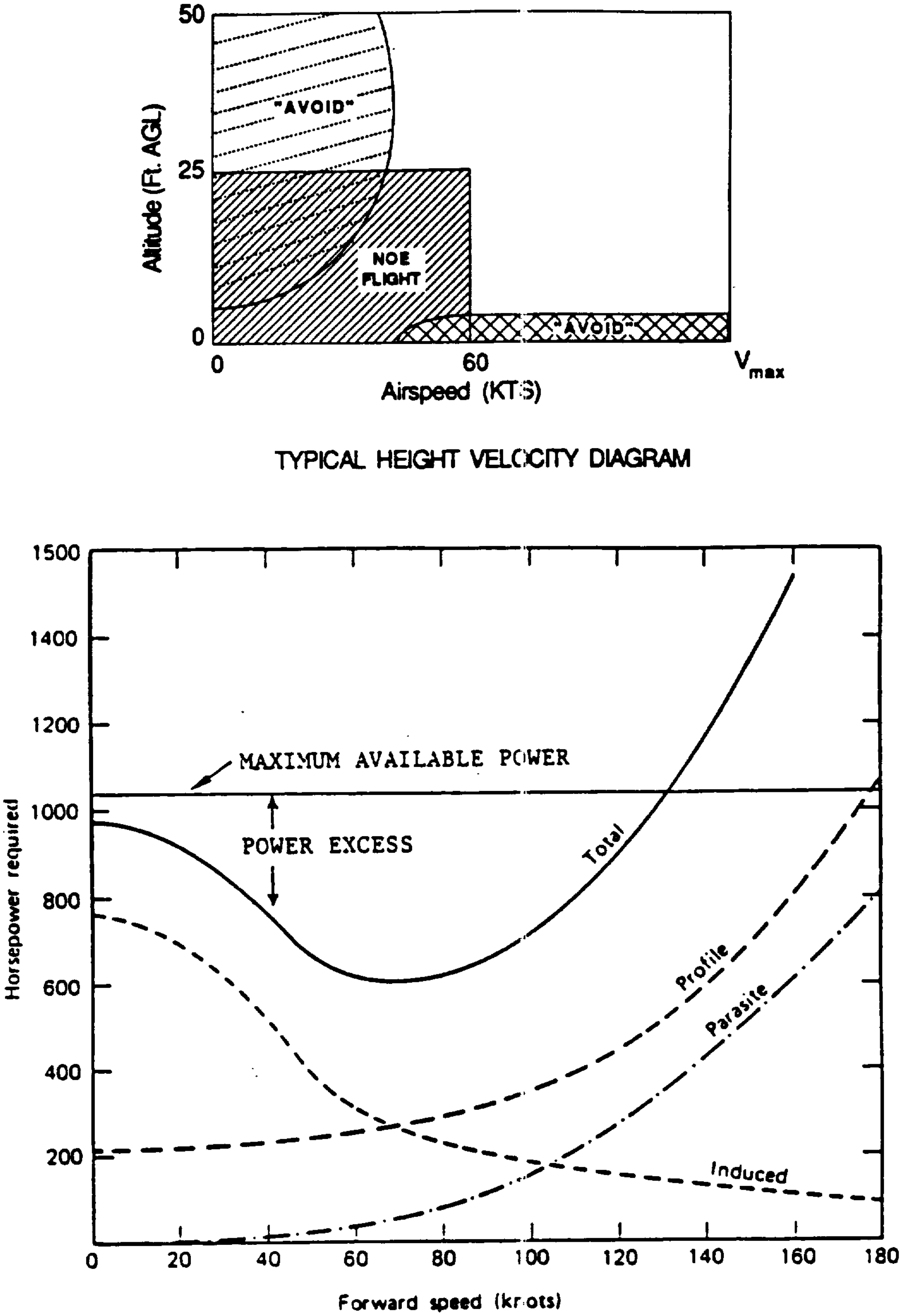
There is a distinct relationship between helicopter flight characteristics and the key sensor requirements to support the concepts of automatic guidance. Since it is the focus of this study to identify and pursue key research areas which advance this technology, this survey of sensing requirements was performed at a level sufficient to define the requirements for generic sensors. The results of this effort clearly indicate the connection between helicopter operating characteristics and the range and field of view/regard of sensors to support these operations.

The principal area of interest in this study surrounds the NOE flight environment. This operating mode is characterized by flight at altitudes of 5-50 feet above the local terrain and at velocities in the range of 0-60 knots. This flight envelope is illustrated in Figure 8 with a typical height-velocity diagram. Turning maneuvers at this low altitude must include enough lift to avoid loss of height above the terrain. Maneuvers, especially at low speeds and around densely distributed obstacles, are frequently uncoordinated and are often typified by slipping or skidding in turns, or consist of pedal turns which generate a change in the tail position with a minimum movement over the ground. Large displacement maneuvers, however, are principally coordinated turns, climbing, or stopping maneuvers. Figure 8 shows the power available and required for a representative light helicopter as a function of velocity in level forward flight. The maneuvering capability of a helicopter is dependent upon the available power beyond the component required to overcome drag and maintain lift. Note, in Figure 8, the available excess power in the NOE velocity range from 0 to 60 knots. This represents the power which is available for maneuvering. The actual power available varies also with aircraft weight, altitude, and temperature.

Table 3 outlines the performance limits for most helicopters and the values which are assumed within this study for NOE flight. The maneuver-relevant parameters of velocity, acceleration, climb, and pitch and bank angles are well within the overall performance envelope. Some parameters, such as max bank angle, are restricted by the velocity dependent power margin as shown in Figure 8. The pitch angles for forward acceleration and deceleration and the attitude control are factors to consider in determining sensor view angle requirements during maneuvering flight.

2.2.1.2 Helicopter Maneuverability

Four types of obstacle avoidance maneuvers were investigated in the analysis of sensor range requirements. These maneuvers involve climbing over obstacles, maneuvering around them, or stopping. The equations of the trajectory paths were determined as a function of helicopter velocity and maneuver parameters (Table 4). For turns, the parameter is bank angle; for climbs, climb rate; and for stopping, the parameter is pitch angle. The turning or stopping accelerations are determined on the basis of no change occurring in altitude. The maneuvering model was evaluated over a range of



TYPICAL HEIGHT VELOCITY DIAGRAM

POWER REQUIRED IN FORWARD FLIGHT

Figure 8. Helicopter Performance Envelope (NOE Flight)

PERFORMANCE ITEM	PERFORMANCE LIMIT	TYPICAL NOE
Velocity	0 - 170 Kts	0 - 40 Kts
Vertical Acceleration	-0.5 to +2.5 G	0.5 to 1.5 G
Max Rate of Climb	3000 FPM	1000 FPM
Max Rate of Descent	3000 FPM	500 FPM
Attitude Control (Hold)	± 2° Pitch/Roll ± 3° Yaw	
Max Pitch Down in Forward Acceleration	30°	15°
Max Pitch Up in Deceleration	45°	30°
Max Angle of Bank	60°	30°
Time to Max Bank	2 sec	1 sec

Table 3. Helicopter Performance Chart

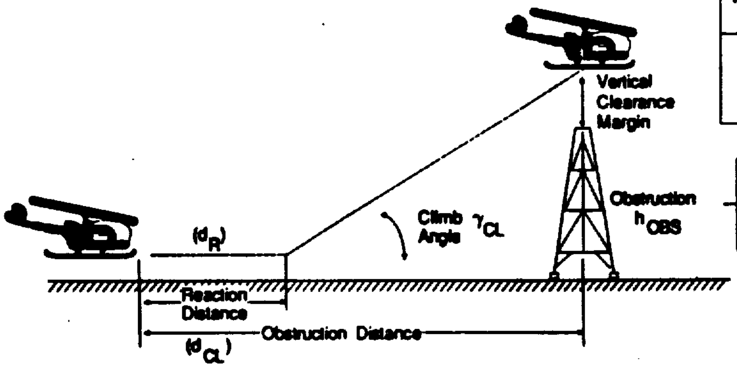
velocities and bank/pitch angles and plotted for use in evaluating the tradeoffs between helicopter performance and sensor range requirements.

Climb Maneuver

Figure 9 illustrates the trajectory for a climb over an obstacle. In this maneuver, the aircraft is assumed to detect an obstacle, perform sensor and data processing functions, and initiate the aircraft response. The time required to perform these functions is the reaction time. The helicopter enters a constant rate-of-climb/constant velocity climb maneuver and clears the obstacle by the same margin as is maintained over the terrain in NOE flight.

CLEARANCE CHART

VELOCITY	CLIMB ANGLE	OBSTRUCTION CLEARANCE DISTANCE
20 Kts	26°	202 Ft
40 Kts	14°	405 Ft



- Reaction Time = 3 seconds
- Climb Rate = 1000 FPM
- Obstruction Height = 50 Ft
- Clearance Margin = Same as Terrain Clearance

Figure 9. Helicopter Vertical Maneuverability

The range equations for the climb maneuver, as listed in Table 4, relate the sensing range to aircraft velocity, climb rate, reaction time, and obstruction height. Two sample evaluations were made for velocities of 20 and 40 knots with an assumed helicopter climb capability of 1000 fpm and an obstacle height of 50 feet. The sensor ranging requirements increase dramatically from 185 feet at 20 knots to 400 feet at 40 knots.

Table 4. Helicopter Vertical Maneuverability

Symbols

v = Helicopter Velocity
v_{CL} = Helicopter Climb Rate
γ_{CL} = Flight Path Angle of Climb
t_R = Reaction Time
h_{OBS} = Obstruction Height
d_R = Distance Traveled During Reaction Time
d_{CL} = Distance Traveled During Climb
d_{OBS} = Sensor Range Required for Vertical Maneuver Over Obstruction

Vertical Maneuver Equations

$$\gamma_{CL} = \tan^{-1} (v_{CL}/v)$$

$$d_R = t_R \cdot v$$

$$d_{CL} = h_{OBS}/\tan(\gamma_{CL})$$

$$d_{OBS} = d_{CL} + d_R$$

Note that this indicates the range at which a reliable range measure is begun. For the passive ranging optical flow system described within this report, additional sensing is required after the aircraft has moved to a closer range.

Quick Stop

Another useful NOE maneuver is the quick stop illustrated in Figure 10. For this maneuver, the helicopter, after sensing, etc., performs a coordinated pitch rotation about the lowest part of the tail rotor. This technique involves a dual process of collective to raise the center of gravity of the aircraft and cyclic to pitch the nose up. Additional collective is demanded to balance the vertical component of the rotor thrust with the aircraft weight. The rearward thrust component provides the deceleration for the maneuver. As the velocity is nulled, either the collective is reduced and the nose is lowered for a hover, or a climb, turn, or other maneuver can be initiated. The equations for the quick stop maneuver are developed (Table 5) with regard to determining the tradeoff between velocity and pitch maneuver and the sensor range required to safely allow the helicopter to be halted. The only other parameter affecting sensing range requirements is reaction time. This item has the same assumption as in the climb maneuver.

Table 5. Helicopter Quick Stop Maneuverability

Symbols

v	= Helicopter Velocity
t_R	= Reaction Time
d_R	= Distance Traveled During Reaction Time
θ_S	= Quick Stop Pitch Angle
a_S	= Quick Stop Acceleration
g	= Gravitational Acceleration Constant
d_S	= Sensor Range Required for Quick Stop Maneuver

Quick Stop Maneuver Equations

$$d_R = t_R \cdot v$$

$$a_S = g \tan\theta_S$$

$$d_S = v^2/(2 \cdot a_S) + d_R$$

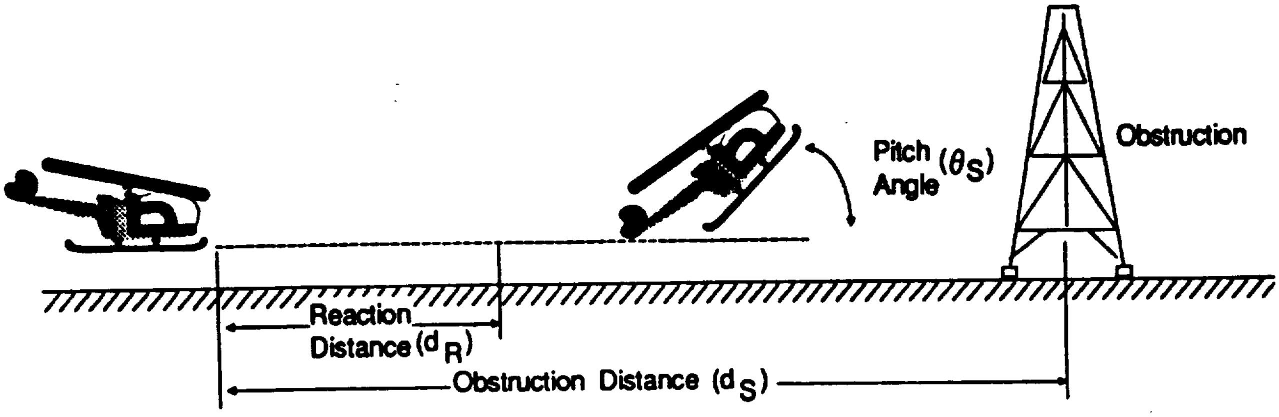


Figure 10. Helicopter Quick Stop Maneuverability

Lateral Turns

Two constant altitude NOE turn maneuvers were evaluated. A swerve, or hard brake maneuver, is a single turn made to avoid an obstacle without regard to the heading of the aircraft after the avoidance maneuver. The two-turn maneuver involves two symmetrical turns, one away from the obstacle and the other a reverse turn to regain the heading change as the obstacle is bypassed. These maneuvers are illustrated in Figure 11. In both types of horizontal maneuvers, the trajectory equations (Table 6) are configured to solve for the downrange distance traveled while reorienting the helicopter to miss the obstacle by a fixed distance or offset. In each maneuver, the reaction time is assumed to include sensor and data processing and the initiation of the bank maneuver for the turn. For either maneuver, the turn radius is a

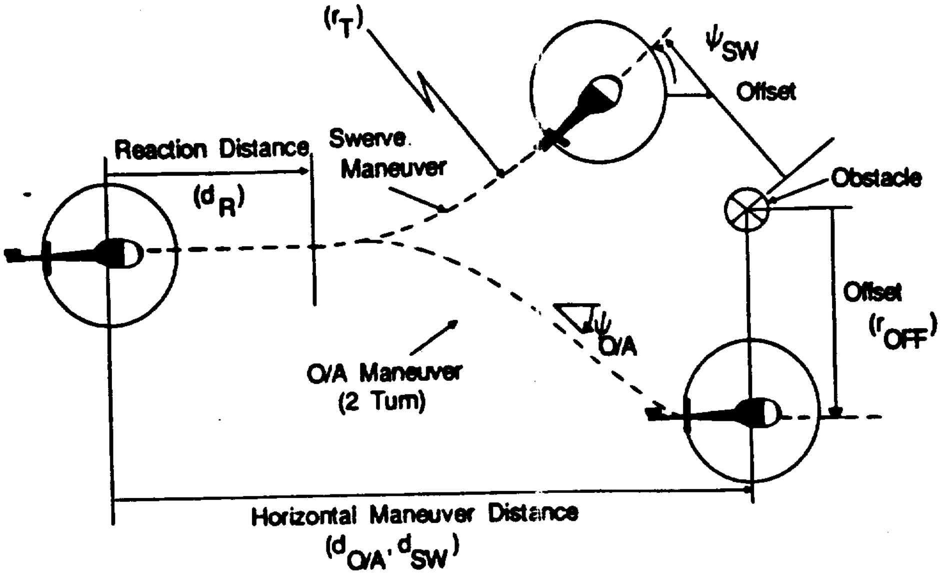


Figure 11. NOE Horizontal Maneuvers for Obstacle Avoidance Swerve and 2-Turn

Table 6. Helicopter NOE Horizontal Maneuvers Swerve and Two-Turn

Symbols

v = Helicopter Velocity

t_R = Reaction Time

d_R = Distance Traveled During Reaction Time

θ_B = Maneuver Bank Angle

r_T = Horizontal Turn Radius

n = Load Factor (Number of g's Experienced)

g = Gravitational Acceleration Constant

r_{OFF} = Centerline Offset Required to Avoid Obstacle

ψ_{SW} = Heading Change During Swerve Maneuver

d_{SW} = Sensor Range Required to Allow for Swerve Maneuver

$\psi_{O/A}$ = Maximum Heading Change During Two-Turn maneuver

$d_{O/A}$ = Sensor Range Required to Allow for Two-Turn Obstacle Avoidance Maneuver.

NOE Horizontal Maneuver Equations

$$n = 1/\cos(\theta_B)$$

$$d_R = t_R \cdot v$$

$$r_T = v^2/(g \tan\theta_B)$$

$$\psi_{SW} = \cos^{-1}(r_T/[r_T+r_{OFF}])$$

$$d_{SW} = (r_T+r_{OFF})\cdot\sin(\psi_{SW}) + dr$$

$$\psi_{O/A} = \cos^{-1}(1 - r_{OFF}/r_T\ /2) \text{ when } r_{OFF}>2r_T, \text{ otherwise } 90°$$

$$d_{O/A} = 2\cdot(r_T+r_{OFF})\cdot\sin(\psi_{O/A}) + d_R$$

function of the velocity and bank angle parameters selected. The trajectory equations do not account for the time required to roll from one bank angle to the opposite. In a nominal 40 knot, 30° bank maneuver, as much as 50 feet of extra range may be required for the two-turn maneuver compared to the swerve maneuver.

In addition to velocity, bank angle, and reaction time, the lateral offset is a parameter which affects the range solution. In this study, the offset was fixed at 50 feet and represents both the allowance for the aircraft rotor and the margin of clearance of the obstacle. Obstacles of appreciable width must also be considered in terms of the offset.

Table 7 lists the solutions for each of the maneuver equations for a parametric variation in velocity. The fixed parameters listed at the top of the table are felt to be representative of NOE flight. A key solution value to note is the greatest sensing range requirement at a 40 knot nominal velocity (418 ft/30° bank) for the two turn maneuver. Note also that at reduced velocities the ranging requirements diminish, but the turn angles increase. It is important for a sensor system to maintain coverage in the intended downrange direction, hence the 20 knot velocity, 56.6° swerve angle is representative of the minimum level of sensor angular coverage a system should have in each direction.

2.2.1.3 Ranging Requirements

Figures 12 through 14 are graphs of the sensor range requirements for a set of velocities and bank/pitch angles. In Figure 12, the velocity sensitivity of sensing range to helicopter velocity is almost linear for all maneuvers except the quick stop. The swerve maneuver tends to require slightly less sensor ranging than the two-turn maneuver and should generally be considered as an occasional tactic to be used when the system is extended beyond the nominal performance. For example, an obstacle which can be avoided with a 30° bank, two-turn maneuver when sensed at 418 ft and at a velocity of 40 knots, can be detected at the same range and velocity and be avoided with a swerve maneuver at a bank of only 20° (Figure 12) or can be detected at 325 feet and be avoided with a 40° banked maneuver.

All maneuvers are heavily driven by the system reaction time. A 3-second reaction accounts for nearly 50% of the sensor range requirement. However, it is unlikely that this time could be reduced by more than one second without employing active sensors and high roll rates.

Figure 13 shows the ranging sensitivity to bank/pitch angle of the helicopter in each obstacle avoidance maneuver at a nominal NOE velocity of 40 knots. A 400-ft sensor range capability appears adequate for a helicopter with 30° bank and pitch capability. However, ranging requirements for the quick stop maneuver grow rapidly if the pitch maneuverability is degraded below 20°.

In Figure 14, the same evaluations are made for a 60 knot velocity helicopter. The sensing range requirements increase to about 600 ft. Note in both Figures 13 and 14 that all maneuvers demand approximately the same ranging capabilities provided in 30° bank/pitch capability and 1000 fpm climb capability is available. Most helicopters have this performance capacity.

FIXED PARAMETERS

Bank Angle (degrees)	30.0	React. Time (sec)	3.0
Rotor Offset (ft)	50.0	G-Load	1.2

VELOCITY (knots)	10.0	20.0	30.0	40.0	50.0	60.0	70.0
Reaction Distance	50.6	101.3	151.9	202.5	253.2	303.8	354.4
Turn Radius	15.3	61.3	137.9	245.2	383.1	551.6	750.8
HORIZONTAL SWERVE MANEUVER							
Swerve Angle	76.4	56.6	42.8	33.8	27.8	23.5	20.4
Swerve (ft)	114.1	194.2	279.5	366.9	455.2	543.9	633.0
TWO-TURN O/A MANEUVER							
O/A Turn Angle	90.0	53.7	35.0	26.1	20.8	17.3	14.8
O/A Dist. (ft)	81.3	200.1	310.3	418.2	525.4	632.2	738.7
VERTICAL MANEUVER (Climb Rate = 1000 fpm)							
Obstacle Ht.	50.0						
Climb Angle	86.8	30.8	19.6	14.4	11.5	9.5	8.1
Climb Distance	2.8	83.8	140.6	194.2	246.5	298.3	349.7
Clearance Dist.	53.4	185.0	292.5	396.7	499.7	602.1	704.1
QUICK STOP MANEUVER (Pitch at 30 degrees)							
Stop Time (sec)	0.9	1.8	2.7	3.6	4.5	5.4	6.4
Distance (ft)	7.7	30.6	69.0	122.6	191.5	275.8	375.4
Total Distance	58.3	131.9	220.9	325.1	444.7	579.6	729.8

CONCLUSIONS

NOMINAL RANGING REQUIREMENTS » 400 FT **MINIMUM FIELD OF REGARD 120 DEG**

Table 7. Helicopter Performance Analysis in NOE Maneuvering for Varying Velocity

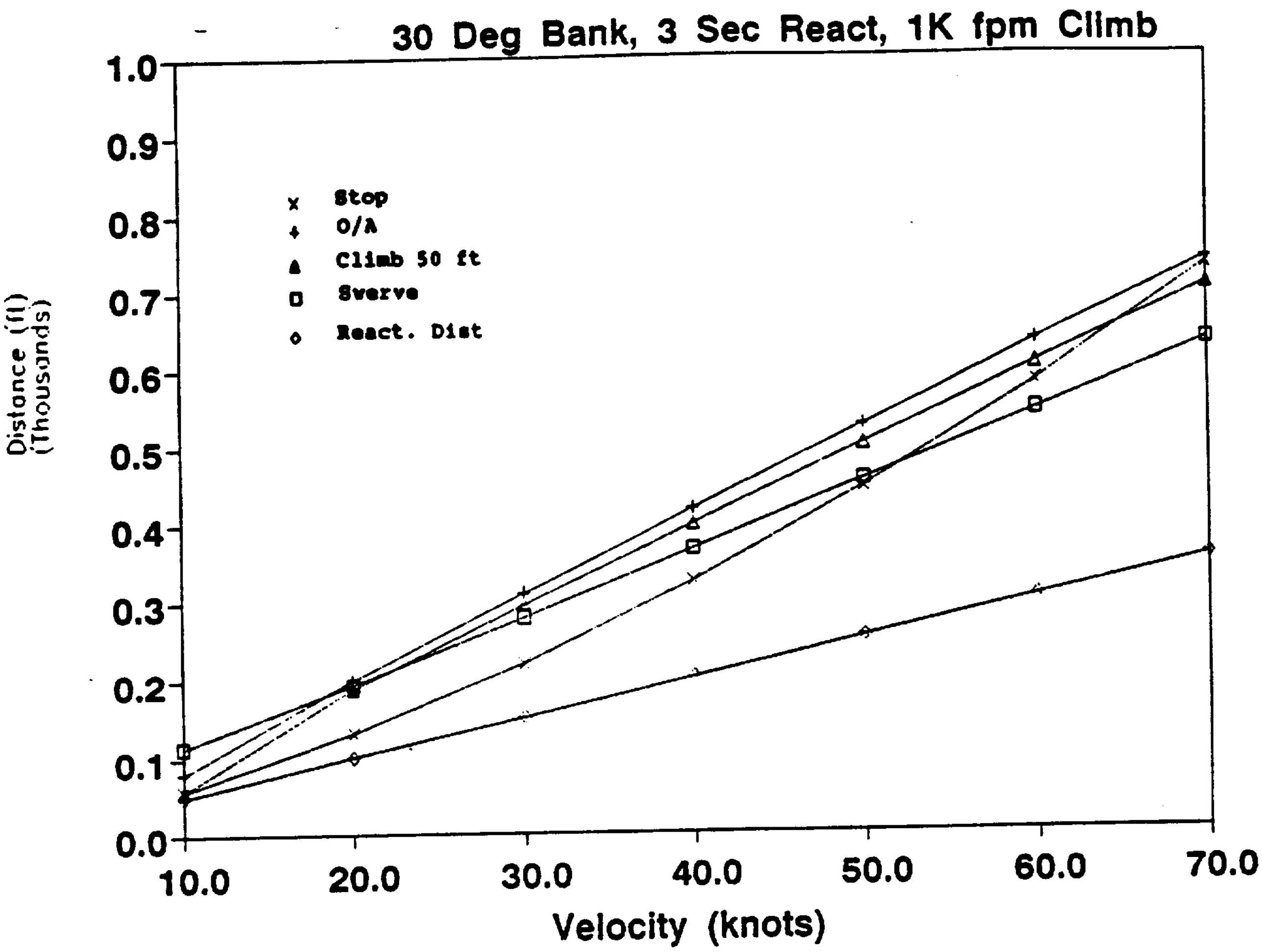


Figure 12. Sensor Range Sensitivity to Velocity

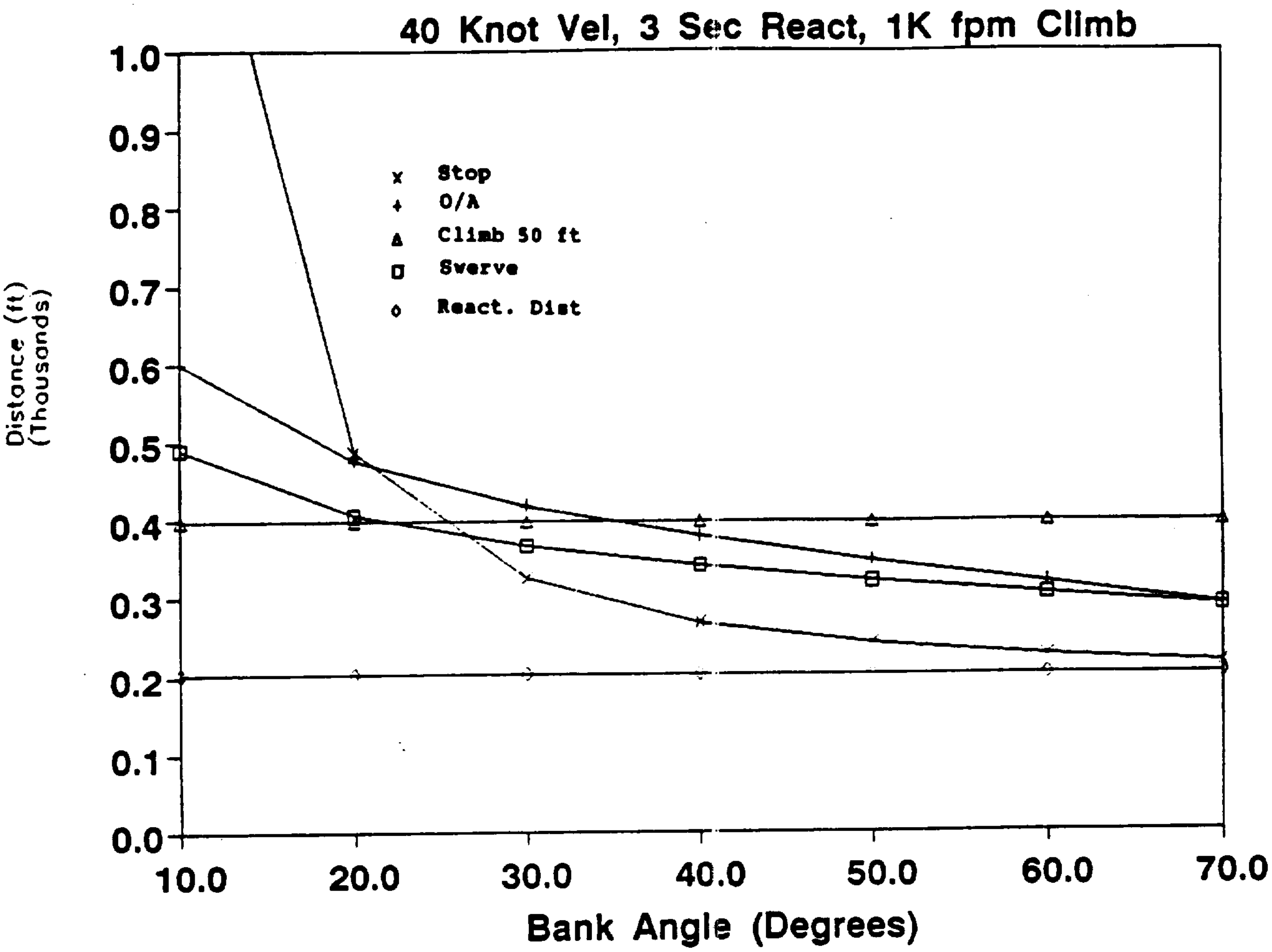


Figure 13. Sensor Range Sensitivity to Bank Angle at 40 Knot Velocity

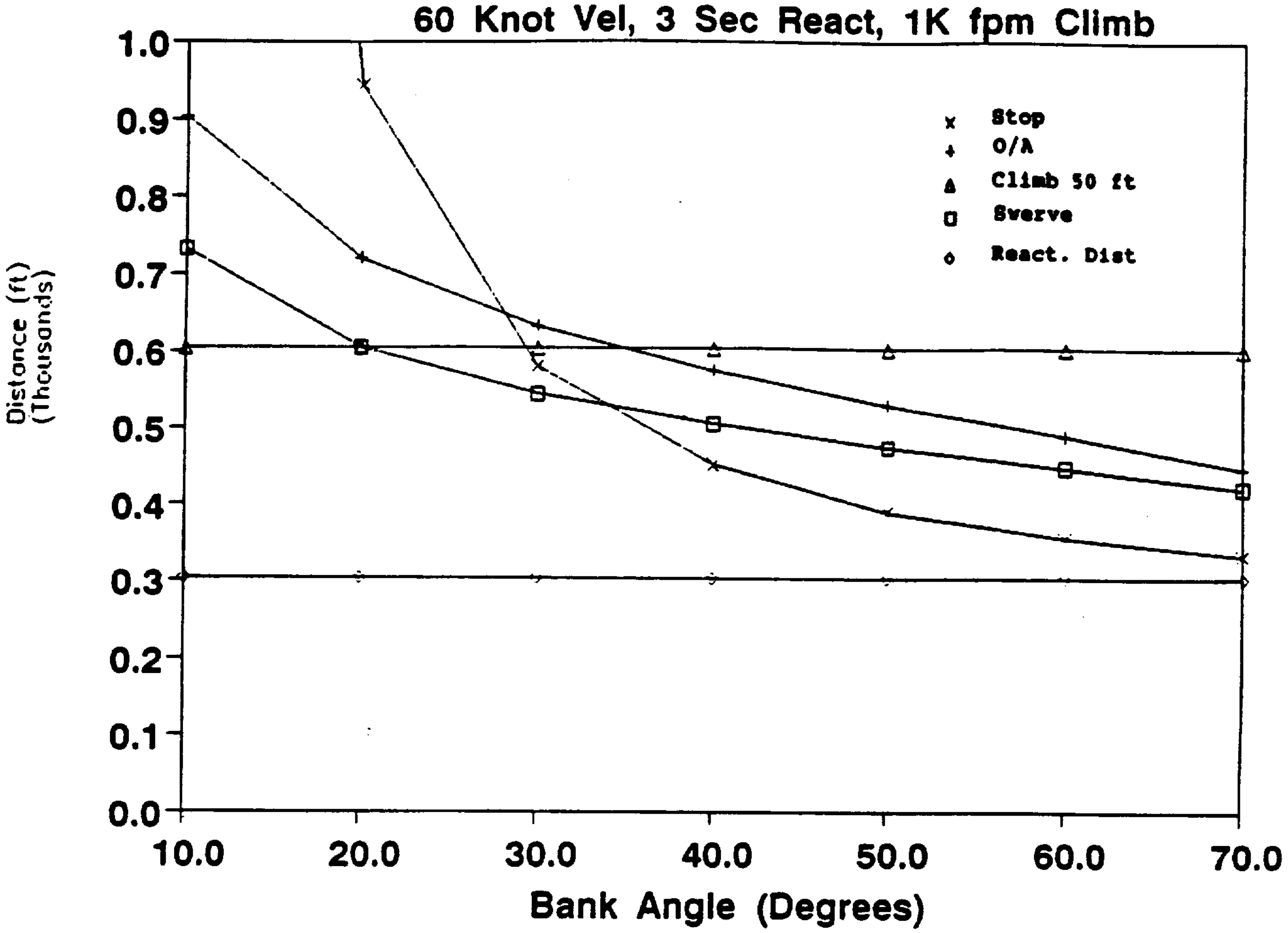


Figure 14. Sensor Range Sensitivity to Bank Angle at 60 Knot Velocity

2.2.1.4 Performance Summary

The performance studies conducted under this study serve merely to outline the approximate requirements for a sensor system. To go beyond the level contained herein would require a detailed definition of a proposed aircraft complete with such details as roll acceleration capability, aircraft control system response, etc.

The performance review strongly indicates that a sensor system should reliably function at ranges of 400-600 ft for a helicopter NOE flight regime of 40 knots and coordinated turn/pitch maneuvers of 30°. Expanding the sensor system performance to ranges of 600-800 ft allows for a blend of higher transit speeds, lower bank angle requirements, and greater obstacle clearance margins.

It should be noted that the system reaction time provides a large contribution to sensor ranging requirements. the reaction time is composed of four elements:

- Sensor Multiple Observations
- Sensor Data Processing and Verification
- Pilot/System Response
- Aircraft Response

The need for multiple sensor observations is discussed in detail in Section 2.2.2 for a passive ranging system. It requires a distinct displacement in helicopter position between sensor observations and, as such, is unlikely to ever require less than approximately one second. Similarly, pilot and aircraft responses are also finite and must include allowances for off-nominal conditions (e.g., loading, etc.).

The actual sensor data processing and verification time requirements are the most likely time components for optimization. Conclusions reached in Section 2.2.2 indicate that emerging technology should be capable of processing data in a fraction of a second.

2.2.2 Passive Ranging from Helicopters Via Optical Flow Measurement

2.2.2.1 Introduction

The pilot of a helicopter close to the ground estimates the range to objects partly by their motion, or optical flow, in his field of view. An automatic system for passive ranging can operate similarly. A conventional TV camera (generating approximately 512 by 512 pixel images 30 times per second) is an adequate sensor for a passive ranging system that could be used for navigation or that detects objects close enough to a helicopter flight path to be threatening.

This section describes and illustrates image processing algorithms that automate passive ranging by optical flow. Section 2.2.2.2 describes the general concept of passive ranging by optical flow. Section 2.2.2.3 presents a brief error analysis that shows that the passive ranging is accurate enough to be useful with practical image measurement errors and navigation system errors. Section 2.2.2.4 describes image processing for pilot warning which uses the concept to detect objects that are close enough to a helicopter flight path to be threatening, i.e., that are within a "tunnel of safe passage" that must be maintained around the helicopter flight path. Section 2.2.2.5 describes other processing techniques that could be used for more accurate obstacle location.

2.2.2.2 General Concept

The speed and direction of the motion, or optical flow, of objects in a pair of images collected from a moving vehicle depend on the

- velocity magnitude
- time between images
- angle of the object away from the velocity vector
- the ranges to the objects.

If the velocity magnitude and bearing is supplied by the vehicle navigation system, and if a feature is detected in two images and its angular motion measured, then the range to the feature can be estimated.

Figure 15 shows how the angle between the feature and the velocity vector might appear in a forward looking camera on the moving vehicle. The image coordinates are azimuth and elevation. The feature displacement in angle from the velocity vector is represented in polar coordinates ρ, θ. (The direction of the velocity vector does not have to lie within the image.)

Figure 16 shows how a motion of the vehicle over distance VT changes the range and bearing to the feature, where r_o and ρ_o are the initial values of r and ρ. The $\omega = \rho - \rho_o$ is the optical flow of the feature.

The mathematical relationships relating the geometric quantities shown in Figures 15 and 16 are*

$$\rho_o = \cos^{-1}[\cos(\alpha_o - \alpha_v)\ \cos(\varepsilon_o - \varepsilon_v)]$$

$$r^2 = (r_o \cos\rho_o - VT)^2 + (r_o \sin\rho_o)^2$$

$$\rho = \sin^{-1}\left[\frac{r_o \sin\rho_o}{r}\right]$$

$$\omega = \rho - \rho_o$$

*The approximate Euler angles are used to describe the general concept. They are not exactly correct for large angles. However, the concept is still valid. An exactly correct formulation could have been given but was not because of its complexity.

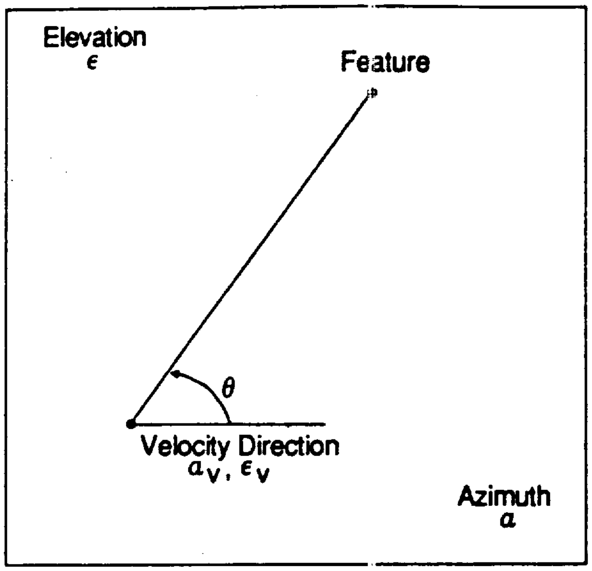


Figure 15. Angles in the Image Plane

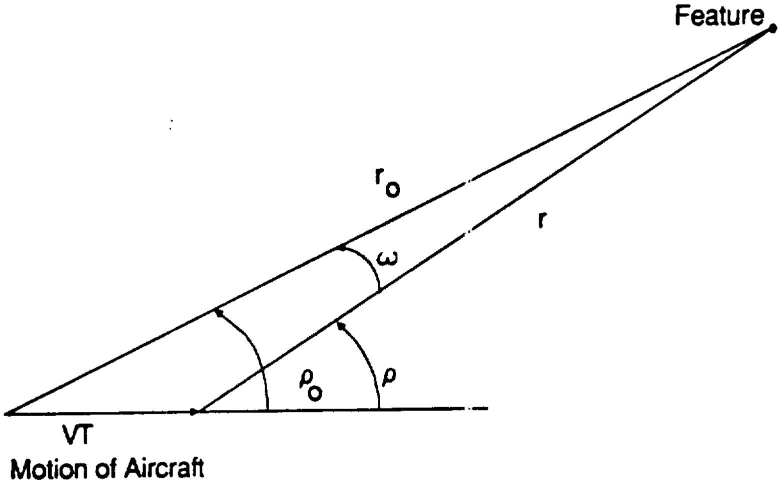


Figure 16. Definition of Range and Angle Away From the Velocity Vector

The equations, if desired, can be solved for the initial range r_o. Figure 17 shows the relationship between the range r_o and the optical flow ω for various angles of the velocity vector, ρ_o. The approximate functional form of these curves is

$$r_o = \frac{(VT)\ \tan \rho_o}{\omega}$$

which is also shown as a dashed line in Figure 17 for $\rho = 20°$.

It should be noted that the suggested technique for estimating passive range from optical flow is much simpler than the general estimation problem because velocity is being supplied by the navigation system instead of also being estimated from the optical flow. The next section will show the sensitivity of ranging error to any errors in the velocity vector that is supplied by the navigation system.

2.2.2.3 Error Analysis

Ranging by optical flow is notorious for being inaccurate. Therefore, an error analysis is in order to shown that the proposed technique is accurate enough to be useful. The standard deviation of the estimated range can be computed using the approximate relationship

$$r_o = \frac{VT \tan \rho_o}{\omega}$$

Differentiating gives

$$\frac{\delta r_o}{\delta \omega} = - \frac{-r_o}{\omega} .$$

Substituting this in the relationship

$$\sigma_{ro} = \left| \frac{\delta_{ro}}{\delta \omega} \right| \sigma_{\omega},$$

gives

$$\frac{\sigma_{ro}}{r_o} = \frac{\sigma_{\omega}}{\omega} ,$$

i.e., errors in estimated range, r_o, are proportional to errors in measured optical flow, ω. The sensitivity of estimated range to other measured or supplied parameters can be computed from the mathematical relationships given previously:

$$\rho_o = \cos^{-1}[\cos(\alpha_o-\alpha_v)\ \cos(\epsilon_o-\epsilon_v)]$$

$$r^2 = (r_o\cos \rho_o-VT)^2 + (r_o\sin \rho_o)^2$$

$$\rho = \sin^{-1} [\frac{r_o\sin \rho_o}{r}] + \rho_d$$

$$\omega = \rho-\rho_o$$

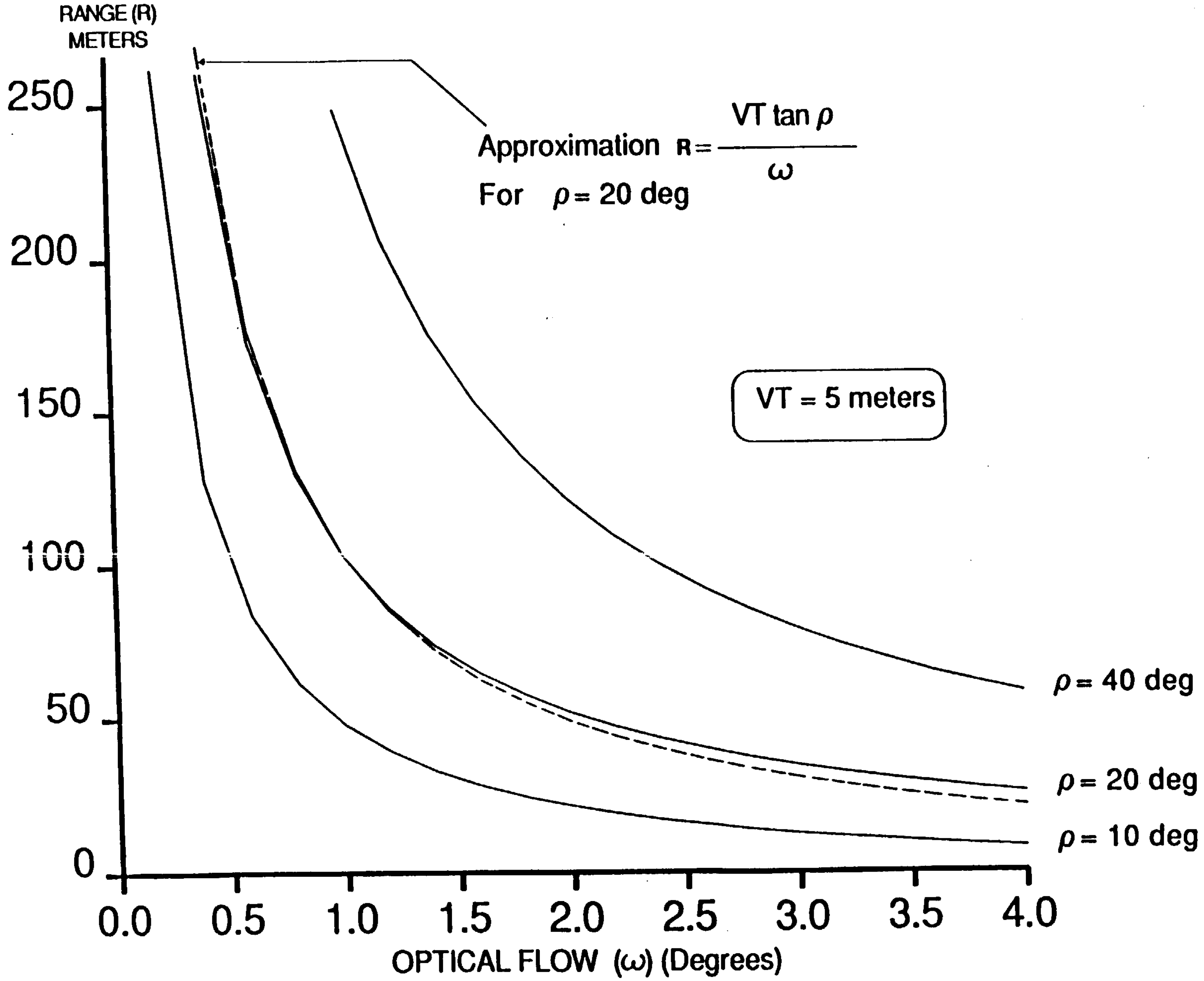


Figure 17. Range as a Function of Optical Flow and Angle Away from the Velocity Vector

where ρ_d is another parameter representing the component iŋ the radial direction of any uncorrected change in the velocity vector.

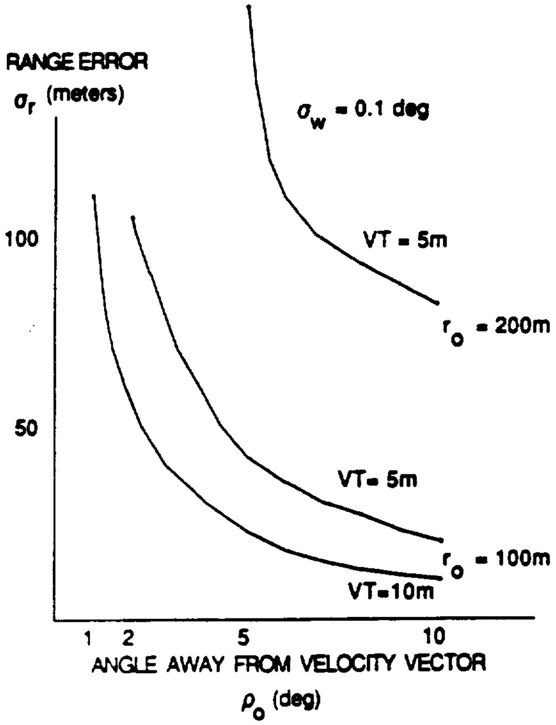
The sensitivity of the estimated range to any parameter p in these equations is given by

$$\frac{\delta r_o}{\delta p} = -\frac{\frac{\delta\omega}{\delta p}}{\frac{\delta\omega}{\delta r_o}},$$

where the partial derivatives of ω can be evaluated using the chain rule to differentiate the above mathematical relationships. For example, using approximate relationship for estimated range

$$\frac{\delta r_o}{\delta p} = -\frac{r_o}{\omega}\frac{\delta\omega}{\delta p}.$$

Typical error in the measured optical flow of a feature might be 0.1°. This approximately corresponds to one pixel in a 512 by 512 image used to cover a 50° field of view. Figure 18 shows the variation of error in estimated range with range and angle of the feature away from the velocity vector. The values of 100 m range and vehicle motion of 5 m (1/6 sec at 30 m/s) are the typical values plotted. The random error for estimated range in this case of 100 meters range is 36 meters. Also plotted are a longer range of 200 m and a longer flight time of 0.333 sec or VT = 10 m. The figure shows a strong dependence on all three parameters. Error is proportional to range squared and inversely proportional to distance travelled, VT.



ORIGINAL PAGE IS OF POOR QUALITY

Figure 18. Range Estimation Error for Optical Flow

The sensitivities of estimated range to errors in the system parameters and typical errors in these parameters are given in Table 8. Multiplying the sensitivity times the typical error gives the ranging error. The table indicates the contributions from all error sources is less than the random error. The sensitivities to error in the elevation angles are zero in this example, because the elevation direction is perpendicular to the direction to the feature.

Errors in two system parameters might be large enough to degrade performance. One is an error in the bearing of the velocity vector. A 1/2 degree error causes approximately a 10 m bias in range. This, however, will not be overly significant if the random errors are several tens of meters. The other questionable system parameter is the change in the direction of the velocity vector in image coordinates caused by flexing of the aircraft/camera mount between imaging times. If this is less than 0.03 deg (1/2 mm in 10 m), then the error in range estimate will be less than 10 m and will not be significant relative to a random error of 36 meters.

The general conclusions from this error analyses are that range estimation using optical flow is feasible assuming the error sources have approximately the magnitude assumed. The most critical assumptions on error sources are:

- A measurement accuracy of 0.1 deg or 1 pixel when measuring the optical flow. This measurement accuracy should be demonstrated by processing of helicopter-collected imagery.

- A velocity bearing error of 1/2 degree, which corresponds to a navigation system drift of 1/2 nm/hr.

- A velocity drift of less than 0.03 deg between images. This number should be verified by further consultation with experts on inertial systems and helicopter mechanical structure.

Figure 19 shows the variation of the range estimation error caused by these sources as a function of the vehicle motion, VT. The effect of measurement accuracy and the velocity drift can be reduced by increasing VT--assuming that the errors do not increase significantly with the increased time interval. The effect of the velocity bearing error is not changed by changing VT. Therefore, the range estimation accuracy is critically dependent on having an adequate navigation system.

2.2.2.4 Tunnel of Safe Passage

A "tunnel of safe passage" for a helicopter is defined as a circular tunnel ahead of the helicopter that has the current velocity vector as its axis and which is free of obstacles. Figure 20 shows a cross section of the tunnel.

Figure 20 also indicates how objects inside the tunnel are detected. At the helicopter (H), the distance to the tunnel wall (HW), is known for all directions. If an object is detected at distance HO, which is less than HW, then the object must be inside the tunnel.

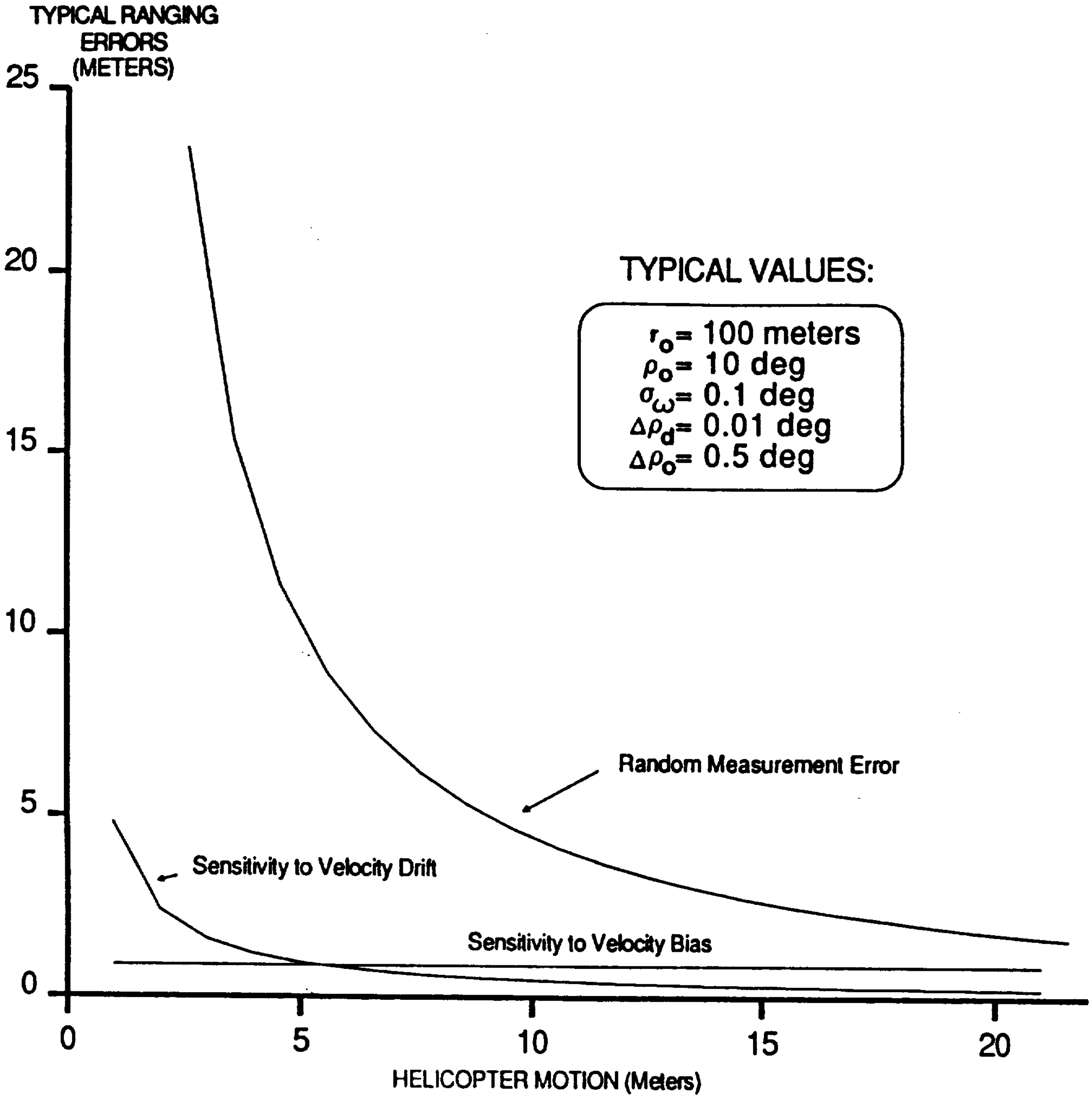


Figure 19. Variation of Range Measurement Error with Length of Vehicle Motion

System Parameter	Typical Parameter Value	Sensitivity	Typical Measurement Error	Typical Range Error
Azimuth to feature	α_o = 5 deg	19 m/deg	Measurement error = 0.1 deg	1.9 m
Elevation to feature	ε_o = 0 deg	0 deg	Same as azimuth	0 m
Velocity azimuth	α_v = 0 deg	-19 m/deg	1/2 nm/hr = 1/2 deg at 30 m/s	9.5 m
Velocity elevation	ε_v = 0 deg	0 deg	Same as azimuth	0 m
Velocity	v = 30 m/sec	3.33 m/m/s	1/2 nm/hr = 1/4 m/s	0.8 m
Measurement interval	T = 1/6 sec	600 m/sec	Negligible	Negligible
Radial drift of velocity direction	ρ_d = 0 deg	362 m/deg	Gyro drift = 0.01 deg Accelerometer drift = 0.01 deg Aircraft flexing = unknown	0.36 m 3.6 m ?

Range to features is 100 meters

Table 8. Error Sensitivity of Estimated Range

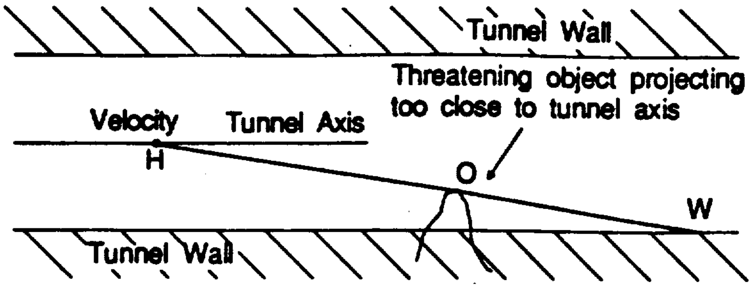


Figure 20. Tunnel of Safe Passage

If a TV camera is imaging in the approximate direction of flight, then the tunnel of safe passage can be superimposed on this image. Figure 21 shows the image of a circular tunnel of safe passage superimposed on the image. The axis of the tunnel is at (V). The rings on the tunnel walls indicate equally spaced ranges from the helicopter.

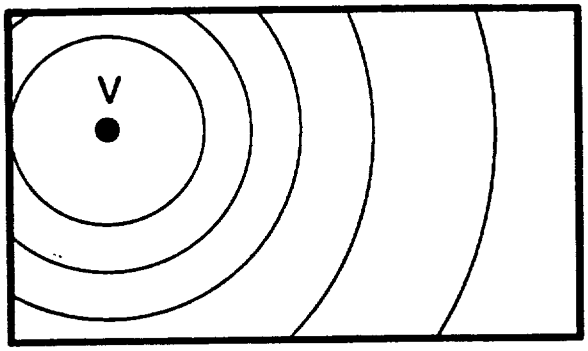


Figure 21. Image of Circular Tunnel with Axis at V

A tunnel of safe passage with parallel walls that extend forever is obviously an approximation. In reality, the helicopter can maneuver and the areas that must be free of obstacles gets smaller with range. However, for this paper, only a circular tunnel of constant diameter will be considered.

For pilot warning, objects within the tunnel of safe passage can be detected without actually estimating their range. Instead, threatening objects are detected because their optical flow is too large. The relationship between object distance from the tunnel axis, and the optical flow, ω, can be derived from the simplified expression

$$r_o = \frac{\tan \rho_o \ (VT)}{\omega}.$$

Realizing that for any object

$$\tan \rho_o \doteq \rho_o \doteq \frac{s}{r_o}$$

$$s = \frac{\rho_o{}^2 (VT)}{\omega} ,$$

where s is the object distance from the tunnel axis. For any s less than the tunnel radius s*, the optical flow, ω, will be greater than ω^* where

$$\omega^* = \frac{\rho_o^2(VT)}{s^*} .$$

Thus, all objects inside the tunnel will have $\omega > \omega^*$, where ω^* is a function of ρ_o, the radial angle of the object away from the velocity direction.

Figures 22 through 25 demonstrate the concept of the tunnel of safe passage for a helicopter. Figure 22 is one of two successive images taken on a low flying helicopter 1/24 sec apart. Figure 23 illustrates the technique for detecting edges. The velocity vector (shown as a cross in the upper left corner of the image) is the center for several radials. The red and blue lines are the high contrast edges computed normal to these radials. The red lines are for the first of the two successive images and the blue lines are for the second. For a specific feature, the optical flow is the distance between the red and blue lines. Figure 24 shows colored warning markers on the features in Figure 23 where the edge was detected in both images and the optical flow was large enough to indicate that the object was within the tunnel of safe passage. Again, the velocity direction is marked with a cross. The green, yellow, and red marks indicate the amount by which the features are within the tunnel--varying from slightly inside to considerably inside. The blue rings in the image indicate the position of the tunnel walls at equally spaced ranges from the helicopter. Figure 25 superimposes the key features of Figure 24 on the second image from the low flying helicopter. This indicates the type of visual cues which might be made available to a pilot.

To be practical, the processing to create a display such as Figure 25 must be done in a fraction of a second. Figure 26 shows a block diagram of the required processing. The module descriptions are as follows:

1. <u>Differentiate the First Image</u> in the radial direction, preserving both the positive and the negative signal.

2. <u>Differentiate the Second Image</u> the same as 1.

3. and 4. <u>Reject Lower Magnitude Peaks</u> that are within radial angle S_{MIN} of a higher magnitude peak. Do this both for positive and for negative going peaks. Typical S_{MIN} is 5.

5. and 6. <u>Reject Low Magnitude Peaks</u> that are below a magnitude threshold.

7. and 8. <u>Reject Small Peaks</u> that have an area of only one pixel.

Operations 9 and 10 are repeated for n=1 to n_{MAX}:

9. <u>Find Positive Peaks with Radial Spacing n</u> between the two images.

10. <u>Find Negative "Peaks" with Radial Spacing n</u> between the two images.

11. <u>Display the Threatening Objects</u> on the second image with the degree of threat.

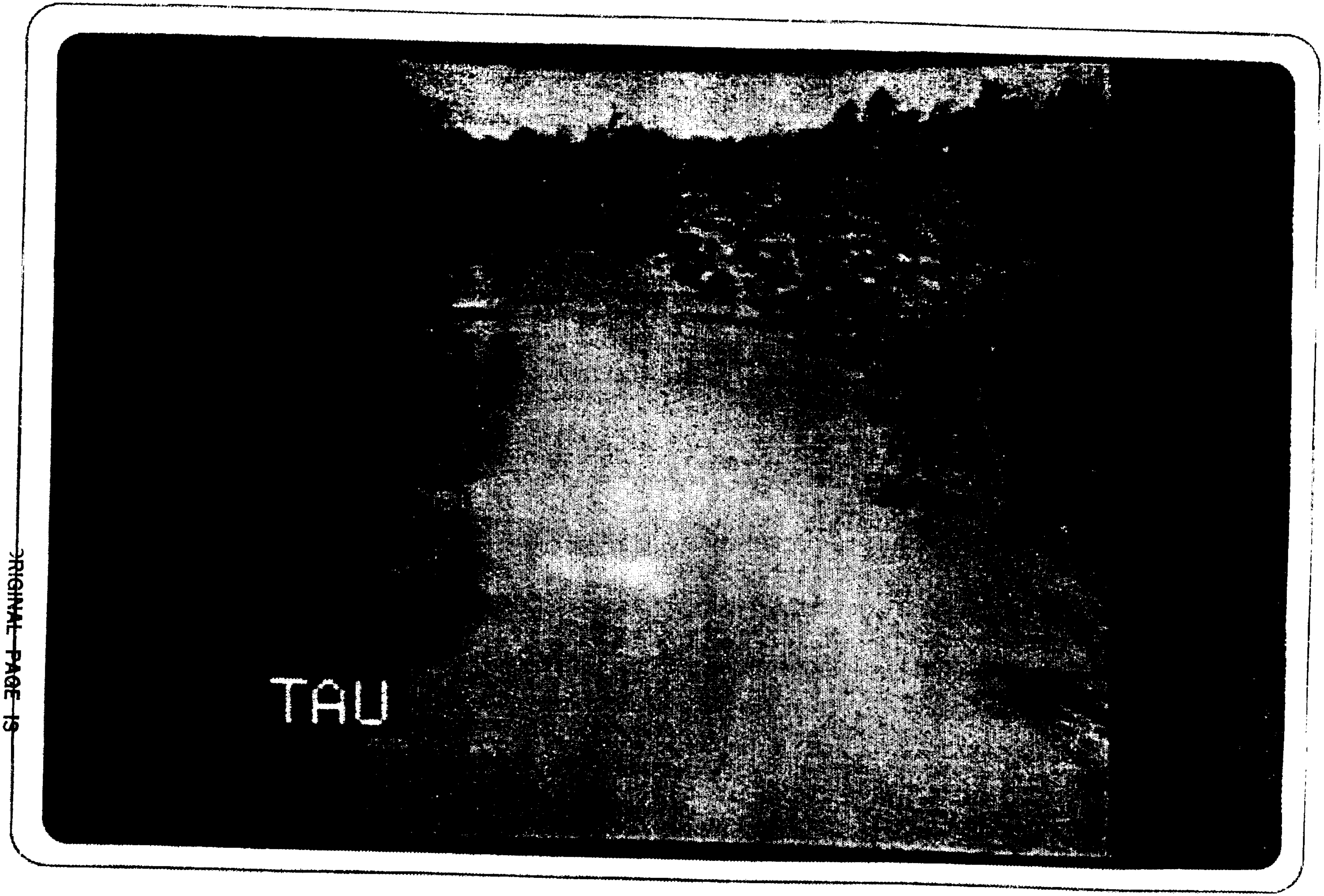


Figure 22. First Image from Low-Flying Helicopter

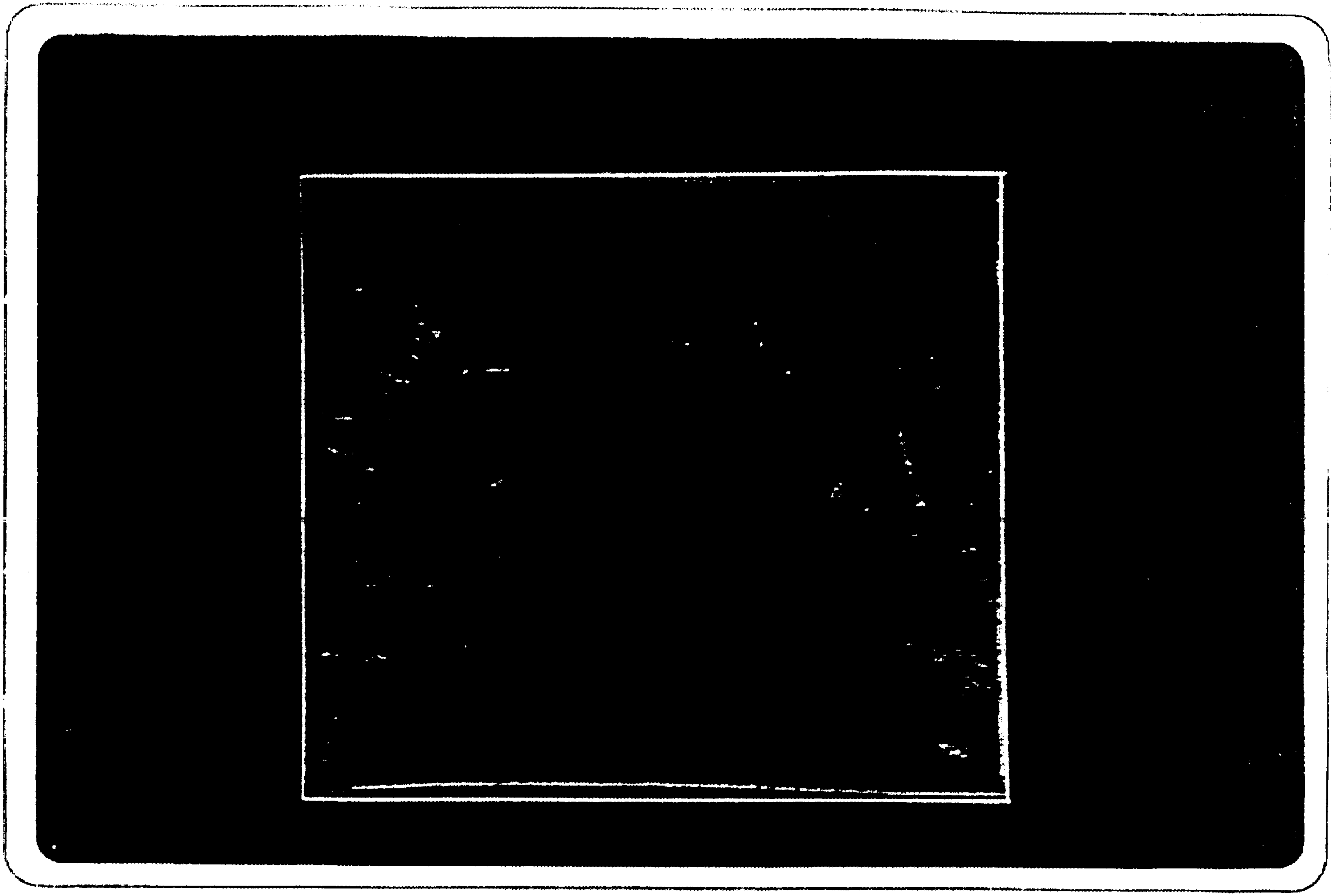


Figure 23. Detected Edges

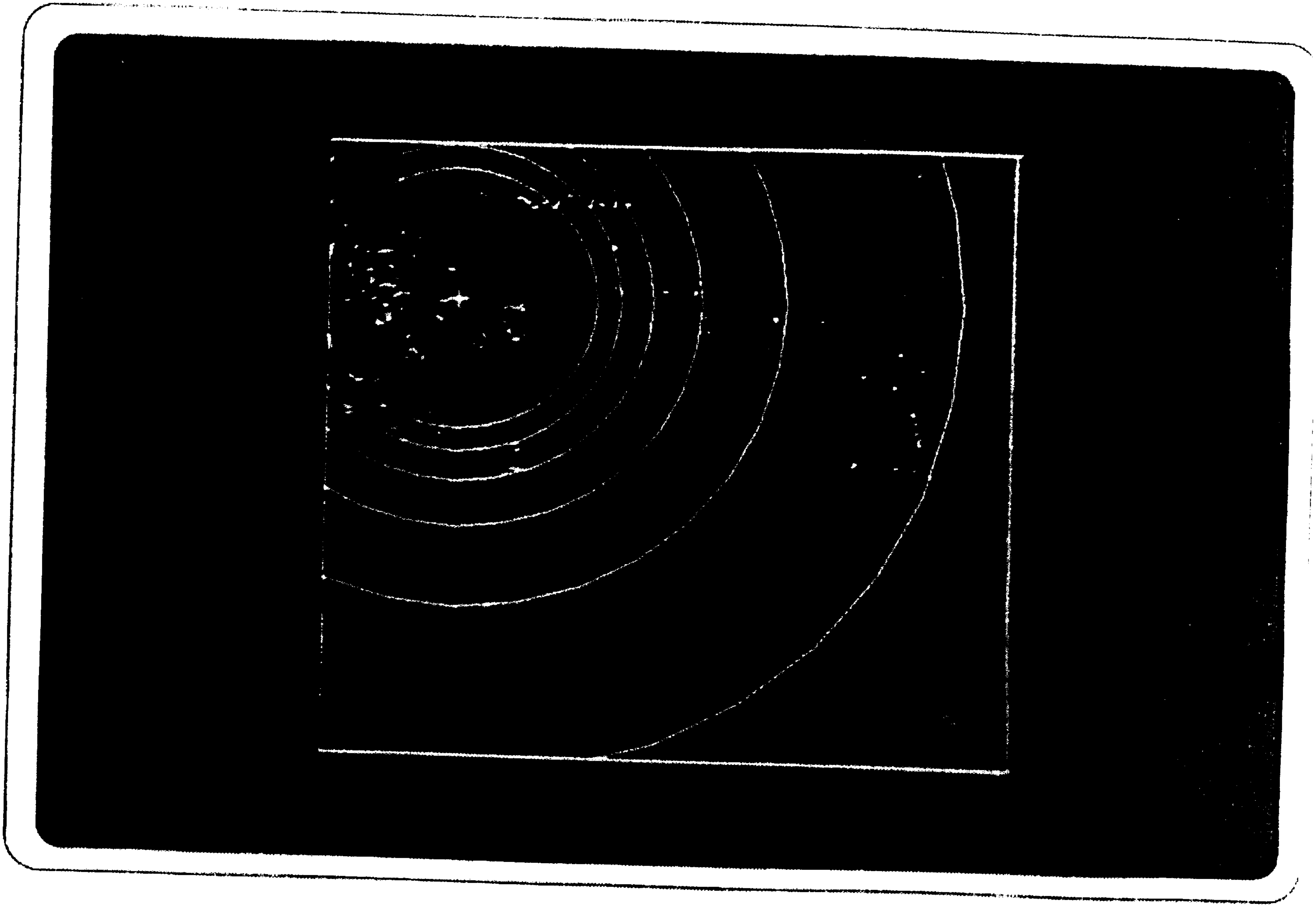
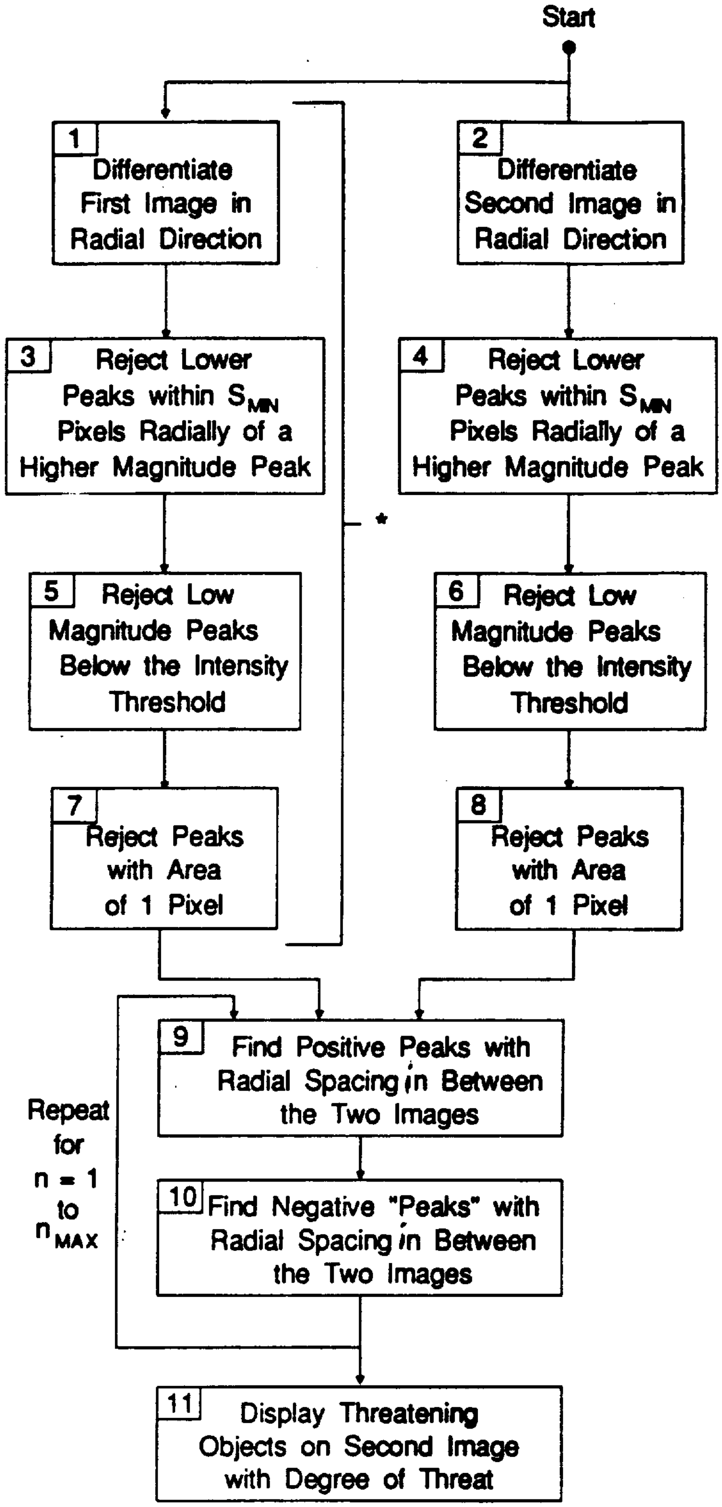


Figure 24. Warning Markers for Threatening Features

Figure 25. Visual Cues Applied to Second Image


Start
1 Differentiate First Image in Radial Direction
2 Differentiate Second Image in Radial Direction
3 Reject Lower Peaks within S_{MIN} Pixels Radially of a Higher Magnitude Peak
4 Reject Lower Peaks within S_{MIN} Pixels Radially of a Higher Magnitude Peak
*
5 Reject Low Magnitude Peaks Below the Intensity Threshold
6 Reject Low Magnitude Peaks Below the Intensity Threshold
7 Reject Peaks with Area of 1 Pixel
8 Reject Peaks with Area of 1 Pixel
Repeat for n = 1 to n_{MAX}
9 Find Positive Peaks with Radial Spacing in Between the Two Images
10 Find Negative "Peaks" with Radial Spacing in Between the Two Images
11 Display Threatening Objects on Second Image with Degree of Threat

* The result of these operations could be taken from the previous Image Pair if the processing is being done continuously.

Figure 26. Block Diagram of Processing To Produce Tunnel of Safe Passage Display

Table 9 shows the essential processing times (extra display, etc. functions are omitted) required to do each operation in frame times, which is the basic subdivision of time for video-based image processors. The times given for the Trapix, which was used to generate Figures 23 and 25, are the minimum times needed for computation, assuming unlimited image memory is available. In actuality, most of the time used to compute these figures was used to repeatedly generate masks defining regions within the image where the optical flow would have a specific magnitude and direction.

The times labeled VLSI pipeline processor are for a special purpose pipeline processor build with VLSI chips. These VLSI chips probably will appear on the market in a few years, or could be developed at reasonable cost. The times for the VLSI implementation for each module assume:

- Operations of the first image have already been done;

- Radial operations can be done in 1 frame time instead of 4 by changing the scroll during a one-frame-time operation. The Trapix requires 4 because it must approximate all radial directions as one of four--0, 45, 90, and 135°--and then use one frame time for each direction;

- The selection of the largest peaks in Operations 3 and 4 can be done in one frame time using VLSI;

- The rejection of all peaks of one-pixel area can be done in one frame time using 3 by 3 convolution;

- The finding of peak spacings in Operations 9 and 10 can be done in one frame time using convolution with a kernel that changes over the image.

All of the assumed pipeline processing is feasible to build, which implies that hardware to produce a tunnel of safe passage display can also be built and at reasonable cost.

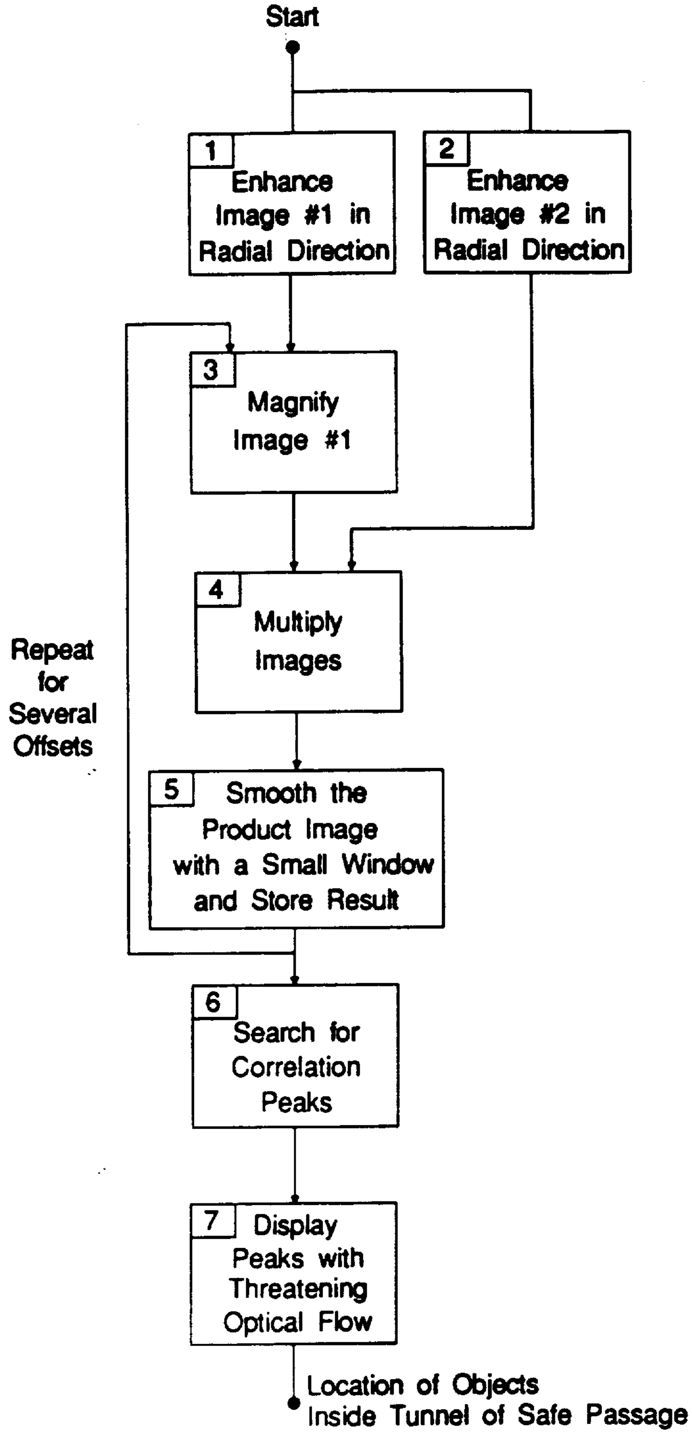

Start
1 Enhance Image #1 in Radial Direction
2 Enhance Image #2 in Radial Direction
3 Magnify Image #1
4 Multiply Images
5 Smooth the Product Image with a Small Window and Store Result
Repeat for Several Offsets
6 Search for Correlation Peaks
7 Display Peaks with Threatening Optical Flow
Location of Objects Inside Tunnel of Safe Passage

Figure 27. Block Diagram of Processing Using Cross Correlation

Table 9. Processing Times To Produce Tunnel of Safe Passage Display

	Trapix	VLSI Pipeline
Operation	(Frame Times)	(Frame Times)
1	4	0*
2	4	1
3	40	0*
4	40	1
5	1	0*
6	1	1
7	9	0*
8	9	1
9	8	1
10	8	1
11	1	1
Total:	125 (4 seconds)	7 (1/4 second)

*The results of these operations can be taken from the previous image pair if the processing is being done continuously.

2.2.2.5 Future Processing Techniques

The optical flow measurement described in the previous section measured the motion of each high contrast feature that was detected in both images. This approach is limited because it can never have accuracy that is much better than one pixel and because false alarms will be generated by falsely detecting an apparent large optical flow that really is made up of detections on two different objects in the two images.

An alternative method of measuring optical flow is to use cross-correlations of small areas in one image with the other image. Because the optical flow at the walls of the tunnel of safe passage is predictable, the number of offsets evaluated in the cross correlation would be small and could be computed rapidly.

Figure 27 is a block diagram of the processing required to compute optical flow using cross correlation. The operation descriptions are as follows:

1. and 2. Enhance contrast changes in each image in the radial direction. The enhanced image from the previous image pair could be used for image #1, if the processing is being continuous.

3. The first image is nonlinearly magnified and converted to polar coordinates about the location velocity direction. The nonlinear magnification is set to equal to the optical flow expected at the walls of the tunnel of safe passage.

Modules 4 and 5 are used repeatedly inside a loop over offsets in ρ that correspond to the larger optical flows expected from features inside the tunnel of safe passage.

4. The images are offset in ρ and multiplied. The offset is approximately equivalent to slightly further changing the optical flow magnification in Module 1.

5. The product image is smoothed with a small window that corresponds to the size area used in the cross correlation. This area will be somewhat larger than the size of the smallest feature expected inside the tunnel of safe passage.

6. Each smoothed product image contains the values of the cross correlation function at one offset for all locations in the image. The different product images are searched for correlation peaks.

7. The locations of the detected correlation peaks are converted to distances inside the tunnel of safe passage and displayed as colored warnings on the sensed image.

Table 9 shows the processing times required to do each module in frame times on the RCI Trapix and on a special purpose processor built with VLSI chips.

The processing times for detecting optical flow given in Table 10 indicate that using cross correlation takes approximately the same amount of processing time as using feature detection. Cross correlation, however, has the potential for subpixel accuracy in measuring optical flow and thus a factor of 3 to 10 improvement in the random error in estimated range. This improved measurement accuracy will improve the detection of objects inside the tunnel of safe passage. This improved accuracy should also make possible other application of passive ranging, such as using multiple rangings to estimate the shape of objects in the image.

Table 10. Processing Times Required To Produce Tunnel of Safe Passage Display Using Cross Correlation

Operation	VLSI Pipeline (Frame Times)
1	0
2	1
3	0 (done as a part of 4)
4	4
5	4
6	0 (done as a part of 5 and 7)
7	1
Total:	10 (1/3 second)

2.2.2.6 Sensor Requirements for Measurement of Optical Flow

A key problem in the study of optical flow for helicopters in an NOE flight environment has been in obtaining suitable data for analysis. It is important to gather real video information rather than to develop a math model of the sensor/obstacle relationship. Real-world data has noise, contrast problems, and unpredictable characteristics which often enhance the researcher's understanding of the operational implications of his ideas.

In our efforts, we borrowed video information from an Autonomous Land Vehicle (ALV) project and substantiated our techniques. Most of the low altitude helicopter flight data was found to be in either a unsuitable format for use on image processing computers or to be cluttered with symbology which rendered the image processing algorithms nearly useless. Finally, a 16 mm noise film was obtained from a flight through the NOE course at Ft. Rucker. This film was converted to video format and used extensively.

The sensor requirements to measure optical flow in helicopter-collected images with enough accuracy to do passive ranging are detailed below:

<u>Video Images/Stability</u>

Detection of optical flow is achieved by detecting changes in two successive images. For this to succeed, the images must be geometrically stable and must have stable gain. Commercial video cameras seem to have the required stability. Many FLIR tapes received in the past have not been usable for optical flow measurements. There have been two problems:

- The horizontal sync has been very poor--probably from video copying, or through possibly the original camera. As a result, a frame to frame comparison could not be done.

- Most FLIR have a very short time constant on their automatic gain control (AGC), e.g., a fraction line scan time. Changes in gain around high contrast objects depend as much on this AGC as on the terrain, making measurement of optical flow difficult. Useful video tapes for any future work in optical flow measurement must have a stable horizontal sync and the normal AGC must be modified. Ideally, the AGC should be "frozen" for the two images being processed for optical flow. In practice the best solution is probably the hold the gain constant for an entire image and then change it only between images.

<u>Navigation Accuracy</u>

The measurement of optical flow must know the location of the velocity direction relative to each of the images (which have azimuth, elevation coordinates). This places two requirements on the navigation system:

- The navigation system must know the bearing of the velocity relative to the aircraft body (GPS is no help here) with an accuracy of approximately 1/2 degree, which for a helicopter with a speed of 60 nm/hr corresponds to an approximately 1/2 nm/hr system.* (The bearing of the camera axis relative to the aircraft body can probably be adequately calibrated.)

- Any change in the velocity bearing between images must be known to 0.03 degrees. Velocity drift of the navigation system is probably not a problem. However, unsensed flexing of the aircraft between the navigation system and the camera might be.

*[(1/2 deg)/57deg/rad)](60nm/hr) = 0.53 nm/hr

2.3 TASK III: INTEGRATED NAVIGATION SENSOR SUITE

2.3.1 Requirements

The need for a robust navigation integration algorithm in advanced helicopters is mission and possibly flight critical. The mission critical need arises from the new mission functions now being planned into such developments as LHX. The navigation suite will support very low level NOE flight. Precise position, velocity, and attitude are required for refined map correlation. Threat localization requires precise position and attitude for sufficient accuracy. No longer can the navigation suite of a modern helicopter be considered a non-critical avionics function. Most major mission phase functions in the mission plan now depend directly on the navigation data for success. Without reliable, precise navigation, the mission is essentially unflyable in terms of survivability and probability of success.

Accuracy requirements for this navigation sensor suite will vary depending on mission segment. Landing requirements are well defined for civil fixed-wing operations by the FAA for low visibility approaches, so provide a representative standard for military rotor-wing operations (although it isn't necessarily true that either military or rotor wing aircraft should be subject to civilian fixed-wing standards). Guidance system accuracy for landing at the most demanding phase (50 foot decision height) requires less than 1 meter, 2 sigma in the vertical axis.

Automated contour flying may be at 5 to 10 feet above the terrain or tree tops. Allowing for guidance system overshoots (undershoots), one must assume that relative ground or obstacle clearance must be, at most, 10% of that distance, or well under 1 foot. Hover hold in the rescue mission, where a line is perhaps lowered through a clearing in trees, has been specified at 1 to 2 meters relative.

These accuracies are certainly demanding; no general geodetic navigation system is capable of such accuracies. Clearly, such relative positioning precision must come from the relative sensors such as radar altimeter, laser rangers, etc., which are capable of accuracies of several centimeters.

Geodetic positioning accuracy requirements derive from any operations that are referenced to a map or other geodetic-based data base. DTED and DFAD data may be assumed, in the general case, to be accurate to 10 meters (although 1 meter data may be available in some areas for DTED). Most military operators have determined their maximum requirement to be 10-20 meters, driven principally by weapon delivery and targeting needs. This is roughly on the level of GPS when operating in the best of conditions. Of course, it is the net performance of the integrated navigation suite that must be evaluated. Evaluation of any individual sensor is not particularly valid; synergistic performance results with deep integration (GPS/IRU, for example), and degraded performance must be addressed, such as under jammed GPS conditions or with a gyro out.

2.3.2 Sensor Architecture

The essential element of any high performance navigation system is the inertial navigation system. The INS is a high rate, reliable, accurate source of velocity change and attitude, therefore serves as the basic platform reference. All other sensors integrated in the system serve to bound its drift characteristics and to aid in failure identification and isolation.

The navigation algorithm integrates the sensor data flowing from the INS and aiding sensors to create a navigation solution at sufficient accuracy and rate to service the various applications and the sensors themselves. The data rates and qualities of the sensors vary widely, from the low-noise, ultra-high rate, reliable signal from the accelerometer to the relatively nonlinear behavior and lagging output of terrain-aided navigation. The output accuracy and rate requirements are at least diverse. In general, the sensor inputs may be locally synchronized, e.g., interior to the INS or GPS, but attempts to assemble the sensor inputs in a global data set will find the sensor data synchronization and time-tagging degrading due to wait states and transit times on the bus.

The output from the INS is relatively well-modelable and linear, with capability for internal fault detection as well as fault isolation given sufficient redundancy. The acceleration and gyro signals are fairly noise-free with well-behaved biases under normal operation that respond well to estimation and correction from external sensor sources. The INS is the primary source of high frequency dynamic data; in coarse terms, the data rate is kept high enough so that the desired dynamics can be found through smoothing. It is also the sensor with highest short-term accuracy and is used to smooth the data emanating from the sensor suite. Many of the navigation applications of the vehicle require output at much higher rate and accuracy than the sensors can provide, for which they rely principally on the capabilities of the INS.

Navigation sensor output varies widely in its accuracy, reliability, and rate, as well as character. The major type of sensor information is position/velocity, used to bound the low-frequency drift of the INS. Velocity information is also available from the sensor suite, varying from very accurate (GPS) to approximate (terrain-aided navigation). Data rates from these sensors are not, in general, adequate to support the higher navigation rates required by the flight, sensor, and weapons control functions of the aircraft and are generally not adequate to support functions such as navigation steering where the update requirements are basically tied to pilot reaction time (about 10 Hz). They can, however, be used to recalibrate the position/velocity states and some of the bias errors of the INS by tracking the cross-correlations via a Kalman filter.

The interplay of the sensors and a simple navigation filter entails an examination of the noise, biases, output delay, and serial error correlation in the measurements. In the context of an operational helicopter, that examination extends to failure modes, failure observability, functional redundancy, and the ability to reconfigure into a different, albeit degraded, fault-tolerant mode after a sensor failure.

The output requirements for the navigation algorithm are, in a sense, dual to the sensor input requirements. Flight and sensor control require a regular stream of estimates of medium accuracy but very high rate. In-flight transfer alignment also requires short bursts of synchronized aircraft state estimates. Similar to certain sensors not being available at all times, these functions may not be required at all times and encourage creating a filter with the ability to reconfigure not only according to the inputs but according to the outputs as well.

2.3.3 Filter Architecture and Partitioning

For simple, well-modeled linear systems, the centralized (monolithic) Kalman filter is inarguably the optimal estimator. As the model increases in size and complexity, however, this strains processing resources. Due to the massive size of the relevant aircraft model and severe constraints on airborne computer throughput capabilities, the mathematical ideal of a complete, centralized Kalman filter has never been realized in the avionics environment. In spite of the fact that throughput capabilities of multiple Common Signal Processing (CSP) architectures (Concept of Air Force Pave Pillar Program) are projected well into the mega-ops range, the centralized filter still cannot be considered as a realistic contender for navigation; for each increase in throughput capability is best absorbed by adding more states to the model in the decentralized filter. Nevertheless, a critical advantage against which all algorithm candidates must be compared is the amount of detailed information that the centralized filter carries. This information can be helpful in performing fault detection and isolation and, in certain formulations, can make the centralized filter particularly easy to reconfigure robustly. Its failing, of course, is its computational burden. In its purest form, it also is inflexible, requiring full filter cycles for even the highest data input/estimate output rates or for parallel filter implementations of multiple model fault detection; the latter case is extremely memory-inefficient as well.

The processing and communication resources dictate the architecture of the navigation algorithm. The problem of estimation in a multi-sensor environment with limited communication between sensors and microprocessors is a distributed estimation problem. A centralized filter will function in such an environment, but with significantly less speed than if it were located in a centralized processor. The set of sensors and associated must constitute a well-matched team in terms of data flow, algorithmic requirements, and processing speed, dedicated to producing both local estimates for the high rate dynamics applications and global estimates for lower rate high accuracy navigation. It is intuitive that the accuracy of the local filter estimates at each sensor and processor will be improved if each one receives appropriate information from the others. In fact, a decentralized filter with sufficient intercommunication between the sensor filters can have optimal linear estimates at each processor. In practice, there are often more efficient ways to use communications capability, and the optimality of the solution is traded off against decreasing information flow requirements.

Because communications between processors is such a critical feature of the filter architecture, efforts must be directed toward determining the relative merits of transmitting raw data (residuals) in addition to the processed data (estimates) between the nodes of the filter structure. Systems requiring more robust estimation generally need a combination of residuals and estimates. Estimates and their associated covariances cannot convey the information necessary to determine whether a filter is diverging from the filter model, so the residuals or a summary of their content must be passed between the component filters for a fuller appreciation of the current dynamic state.

In general terms, the successful distributed filter will have a mix of raw and filtered measurement sharing. The raw measurements are necessary for robust distributed filtering but are associated with high data communication rates. The filtered measurements contain reduced information, only relevant in the context of the filter model which produced, and are associated with, lower interprocessor communication rates. A balance, which would contain only the information relevant in the context of both source and destination filters, such as a summary of the residual content, is necessary.

2.3.4 Integrated Navigation Sensors

Of course, the possible designs which satisfy these general guidelines are many. Final system design optimization requires a precise description of the sensors to be used, their error characteristics, their update rate and precision, their timing synchronization accuracy, their failure modes, and their data output capability. Also, data bus capacity, processor architecture and availability, and sensor/processor/antenna location are necessary. Although a navigation processing algorithm can be generalized to some extent, it can never be completely generalized as long as these constraints exists.

To illustrate a potential navigation system design, we will consider an example of typical advanced helicopter sensors and a representative processing architecture. The sensors to be considered include:

- Dual RLG IRU
- 6-Channel GPS
- Terrain-Following Radar
- CO_2 Laser Obstacle Detector
- Forward-Looking Infrared (FLIR)
- Central Air Data Computer (CADC) (providing primarily barometric altitude)

A first design premise is based on consideration of the relative quality of each sensor assuming that all sensors are fully operational. As stated earlier, none of these sensors can approach the short term precision of the IRU. Similarly, none can approach the long term accuracy of GPS. It stands to reason, then, that for general geodetic positioning and velocity information, the IRU/GPS combination can practically stand-alone as the navigation sensor.

Geodetic position may not suffice for low level operations since the relative navigation requirement of terrain/obstacle avoidance would depend on map accuracy and detail, neither of which are dependable, or even available, to the level necessary to provide safety of flight. For this function, near-field relative navigation sensors such as FLIR, TF Radar, and CO2 Laser must be used. These sensors are a critical component for the relative navigation problem of NOE flight.

If we relax the assumption of all sensors fully operational, the problem grows quickly in complexity. The various sensors now play two roles: system fault detection and isolation, and backup navigation once the culprit sensor has been compensated or taken off line. Even though certain relatively low-quality sensors may be ignored in the centralized navigation solution when all systems, for example, the GPS and IRUs, are operational, they will still be used continuously for the purpose of fault detections. If a primary sensor is taken off line or degrades, the backup sensors may be used for primary position and velocity determinations. The ensuing degraded accuracy capability, although the best now possible, would be the basis for restricting certain flight regimes, maneuvers, or mission functions that require higher performance. In addition, lack of further backup (fail-operative/fail-operative) at this juncture may require similar restrictions due to the inability to recover from the next failure.

The roles of the various sensors for these functions are summarized in Table 11. This table lists the general utility of each sensor based on its performance characteristics. Uses for primary geodetic navigation, NOE relative navigation, and fault detection/isolation are documented.

2.3.5 Integrated Navigation System Concept

As defined in the previous section, the sensors lend themselves in varying degrees of capability to the functions of primary navigation and NOE navigation. In addition, these two functions are quite distinct in their own criticality, computation, and mission use. Therefore, as long as a distributed filtering architecture can be defined that meets the requirements of near global optimality and system-wide fault detection, it makes sense to differentiate these two functions in the architecture. This partitioning is illustrated in Figure 28. The scheme is approximate; clearly there are diverse functions within each of the major filter blocks which may dictate further federation, and other sub-filters may be defined between certain sensors. Note that only one IRU is used as a continuous source of navigation data, but both are calibrated and the second has an immediate fault detection role through parity vector techniques.

Obviously, this design is at a very high level. But it does serve the purpose of illustrating a possible way to federate the common processes within the overall navigation architecture to achieve the purposes of functional partitioning as well as system-wide integration and processing efficiency through distributed processing. Incidentally, there is an advantage to making the federated filters be physically significant filters; the system reconfiguration logic follows naturally, and system developmental testing as well as operational checkout is greatly simplified.

SENSOR	PRIMARY NAVIGATION	NAP-OF-EARTH NAVIGATION	FAULT DETECTION/ ISOLATION	TYPICAL STAND-ALONE ACCURACY
DUAL IRU	● High-rate velocity change ● High-precision ● Attitude reference ● Highest lead derivative ● GPS code loop BW reduction ● GPS state dynamics modeling ● Smoothing	● Geodetic positioning with map	● Internal fault detection/ isolation/recovery ● Other sensor short-term jump or high rate drift	● 0.25 nm per hour
6-CHANNEL GPS	● Low rate, accurate position ● Moderate rate, accurate velocity ● IRU bias calibration ● Altimeter bias calibration	● Geodetic positioning with map ● Absolute/relative frame resolution	● Internal satellite anomaly isolation (6-in-view) ● IRU element failure ● Long-term sensor failures	● 15 m SEP 0.04 m/s
TERRAIN FOLLOWING RADAR	● Moderate rate velocity ● Relative height (AGL)	● Terrain correlation with map ● Terrain avoidance	● Backup IRU velocity check	● Several cm
CO_2 LASER	● None	● Obstacle detection, relative velocity	● Possible backup velocity check	● 1 cm
FLIR	● None	● Low level terrain/obstacle avoidance ● Low level map-making	● Possible velocity source with passive ranging	● 5-10 m/sec
CADC	● Vertical axis smoothing	● Backup altitude source	● Vertical axis fault detection	● 30 m height 1 m/s vert. veloc. (calibrated)

Table 11. Advanced Helicopter Navigation Suite Sensor Roles

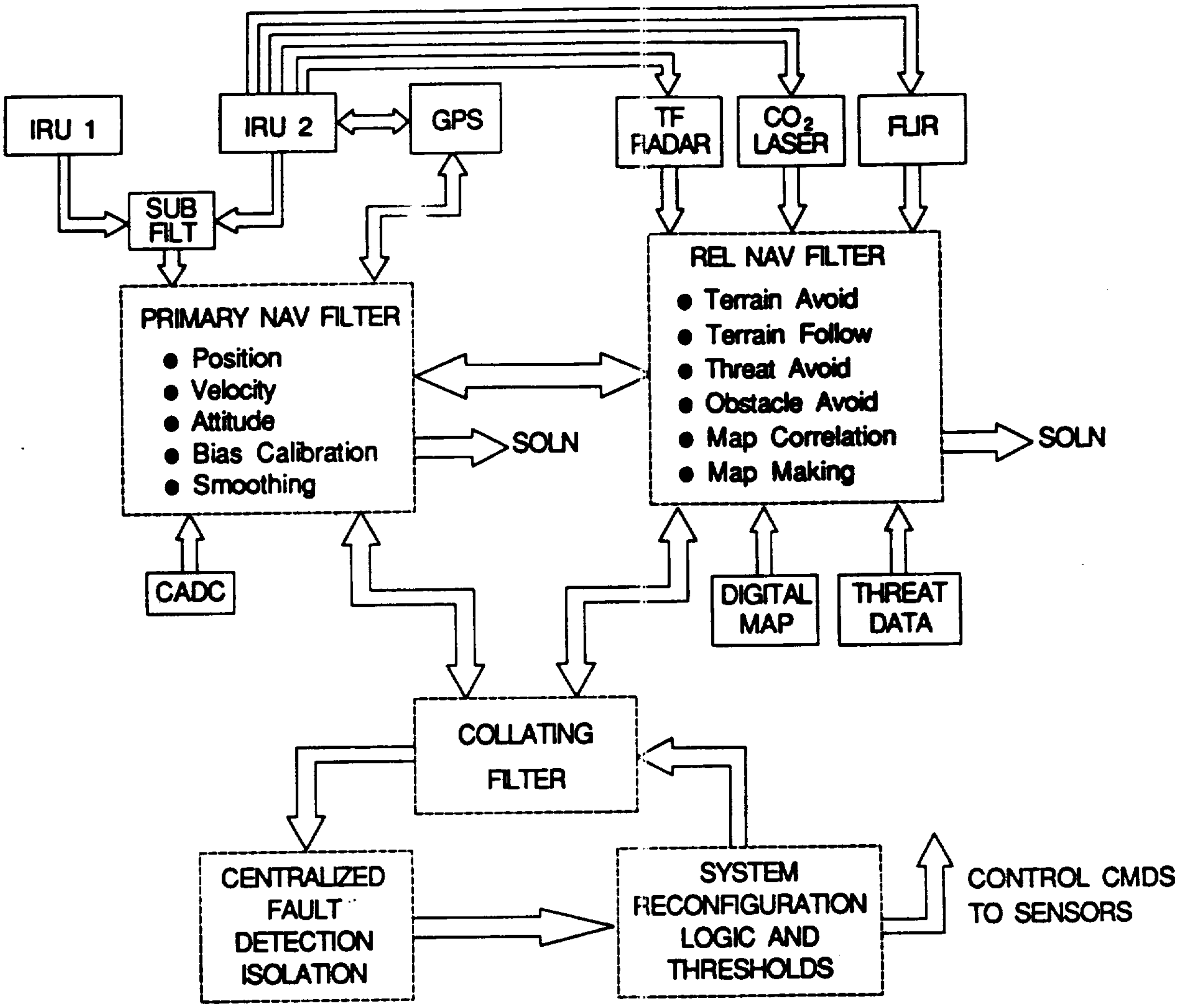


Figure 28. Navigation Sensor Partitioning

2.4 TASK IV: GUIDANCE AND CONTROL LAW DEVELOPMENT

2.4.1 Near Field Navigation

2.4.1.1 Introduction

One of the major efforts of the Autoguidance project was to develop the technology of near field navigation. The near field flight domain requires solving for optimal or near optimal flight path and the consequent aircraft control commands for approximately the next 30 seconds of flight. This time interval is of particular concern because it will typically be using a fusion of long term information (digital map data, known threats, waypoints, etc.) and near term data supplied by on-board sensors (pop-up threats, unmapped hazards, and cultural features). Trajectories must be determined which minimize the risks of exposure and collision, are optimized with respect to the destination (time, fuel, etc.), and are flyable by the aircraft and pilot.

Development of an extensive set of algorithms, Dynapath, is one of several potential technologies which have application in the TF/TA/NOE flight environment. Dynapath has been proven, in principle, to be an optimal control approach that can be implemented in real time.

During the contract period, a significant level of effort was expended in converting the Dynapath code to the NASA-NOE environment. In its initial form, Dynapath was developed under both TAU IR&D and Air Force contract funding as a proof of principle for a viable real-time TF/TA approach for application to high performance aircraft with 6g and 75° bank angle capabilities. Ride quality was not a major consideration in the initial assembly of the software and the trajectories generated for the initial tests had a vigorous behavior. For the NASA applications, a more constrained vehicle performance envelope (typically, .25-2g and 15°-60° bank angle) tends to limit the flexibility in generating paths which react to near term obstacles.

Additionally, the early Dynapath software was benchmarked for real-time operation on a SEL computer. This particular machine runs at about 4 times the speed of the VAX computers proposed for use in NASA-NOE real-time simulations. The Dynapath code computed 30 second patches, or path predictions, in about 5 seconds of machine time.

Although the machine was tasked to perform other operations besides Dynapath code execution, it was apparent that the conversion to the NASA machines would require enhancements to further the speed of execution.

2.4.1.2 Overview of TF/TA Optimization

Before describing the modifications made to the Dynapath software, it may be useful to provide a functional description of the trajectory optimization process and to describe the components and interfaces of the Dynapath algorithms.

In a mathematical sense, the definition of the near-field navigational problem is to find the 3-D trajectory in inertial coordinates which corresponds to a minimum of an optimization performance measure. The trajectory is subject to the following conditions:

- the initial boundary conditions and velocity vector are given;
- the final boundary conditions may be relatively unconstrained;
- the helicopter equations of motion must be satisfied;
- the trajectory must satisfy a range of parameters such as terrain clearance, both laterally and vertically, flight path angle, maximum bank angle, and total acceleration (g-load).

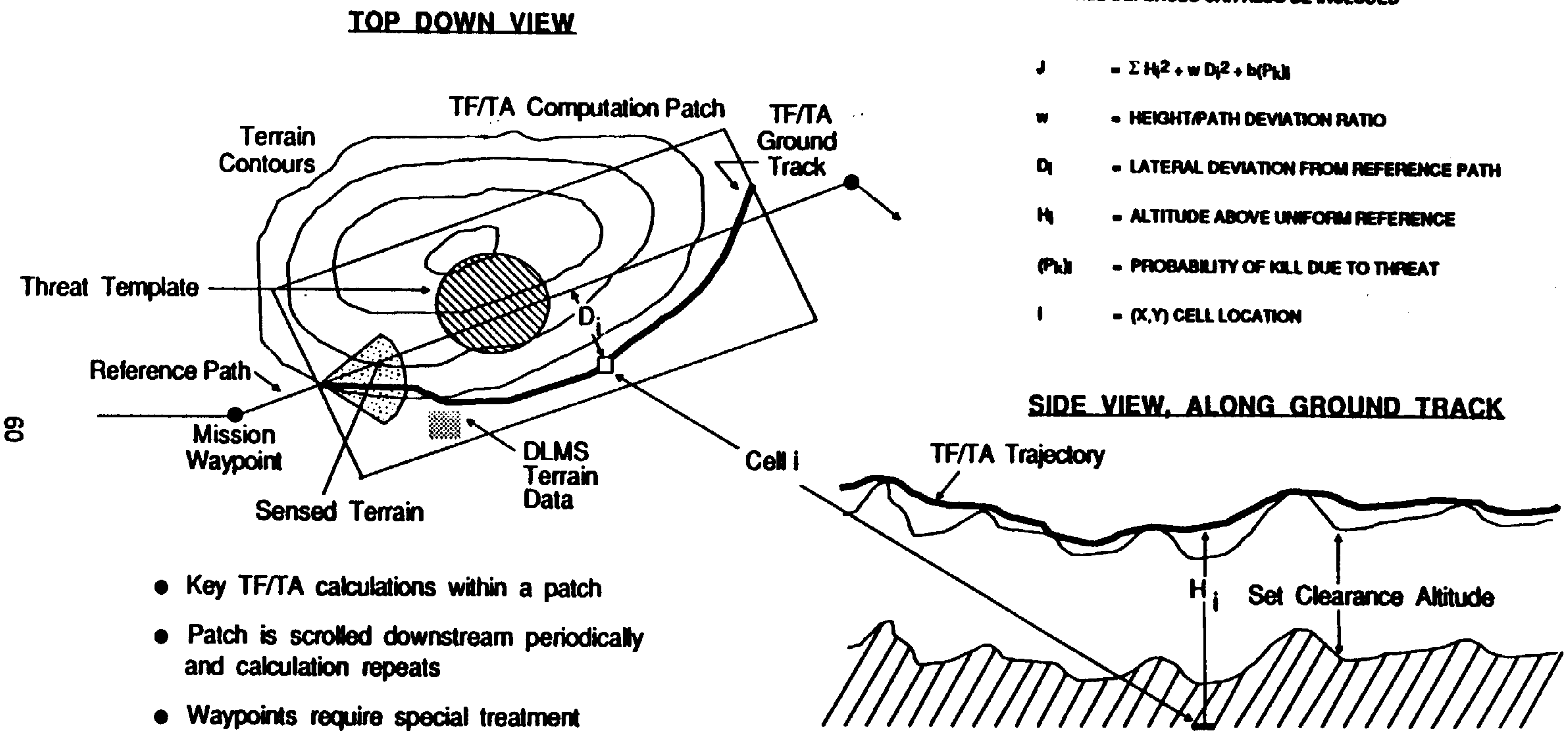
Furthermore, the solution trajectory must have the following features. It should be globally optimal to satisfy the tactical flight objectives. The individual trajectory patches must have an adequate look-ahead to avoid major obstacles. For example, box canyons should be seen within a single patch computation. Additionally, unavoidable ridgelines require sufficiently early detection to initiate a climb rate within the aircraft limits.

Other operational features important to the solution are that the trajectory segments maintain continuity through the first derivative as a minimum. Step changes in the velocity vector are obviously unflyable and bear no approximation to any aircraft capabilities. Continuity of the acceleration profile guarantees an even closer approximation to the performance of an aircraft. For example, a helicopter in a maximum banked turn to the left, cannot immediately reverse itself and turn to the right. It is limited by its roll acceleration capability and, in an NOE environment, the need to maintain sufficient lift to avoid critical loss of altitude. To the same extent, then, the optimization process should generate trajectories which are fully compatible with the flight control system. In general, it has been found that the trajectory and control settings should be provided to a resolution of one second. This time scale is of the order of pilot and aircraft/control response.

Another feature of the solution process required for eventual successful implementation in a flight system is that the method lend itself to real-time operation. The algorithms guarantee a solution within a predictable time.

2.4.1.3 Performance Measure

The TF/TA trajectory computation process is based on optimal control techniques. As in all such approaches, it is necessary to first define a performance measure, or cost functional, against which possible trajectories are ranked and selected. Whereas the global trajectory relates to higher level mission goals, the objective for the real-time trajectory computation is more microscopic or near-term in nature. The TF/TA valley seeking performance measure used in the Dynapath algorithm is shown in Figure 29. This measure uses the global trajectory as a baseline for developing the fine-tuned trajectory, in that lateral deviations from a global trajectory are penalized, while flight at higher altitudes is also penalized. In evaluating all possible trajectories using this penalty function, the best trajectory generally seeks out low altitude corridors ("valleys") in the neighborhood of the global reference trajectory. The relative weight between these penalties



PERFORMANCE MEASURE DEFINITION:
HOSTILE DEFENSES CAN ALSO BE INCLUDED

J	$= \Sigma H_i^2 + w D_i^2 + b(P_k)_i$
w	= HEIGHT/PATH DEVIATION RATIO
D_i	= LATERAL DEVIATION FROM REFERENCE PATH
H_i	= ALTITUDE ABOVE UNIFORM REFERENCE
$(P_k)_i$	= PROBABILITY OF KILL DUE TO THREAT
i	= (X,Y) CELL LOCATION

- Key TF/TA calculations within a patch
- Patch is scrolled downstream periodically and calculation repeats
- Waypoints require special treatment

Figure 29. Patch Computation Performance Measure

is called the TF/TA ratio. A large value for this ratio results in essentially TF flight along the reference trajectory, thus bypassing low altitude corridors, while a small value would permit large deviations (TA flight) in the search for low altitude corridors.

The general philosophy behind this performance measure is that low altitude corridors afford terrain masking from threats, and thus represent good candidates for improvement over the global reference trajectory. However, recent testing [Ref. 8] has shown that threats and terrain masking should also be incorporated explicitly for best performance. Otherwise the TF/TA trajectory may go through a threat region unnecessarily. Mathematically, inclusion of threats can be achieved by adding to the TA/TA performance measure a term $\beta\ (P_k)_i$, associated with the threat danger P_k in cell i.

It is worth noting that the TF/TA ratio ω should, in principle, be adjusted from one patch to the next. The ratio serves as a proxy for the appropriate trajectory adjustments to account for threats. However, such adjustments are a complex function of the threat laydown, of clobber, and other effects, as addressed by a global trajectory generator. Explicit inclusion of a threat penalty term in the TF/TA performance measure builds additional "intelligence" into this performance measure and would reduce somewhat a need for careful adjustments to ω. In any event, the ratio ω is simply treated as a constant parameter in the present study.

The optimum trajectory is determined by summing the incremental costs associated with each step, or time interval, in the trajectory. The connected set of steps with the minimum total cost is the optimum. These incremental costs are referred to as "$cst(x,y,\psi,\rho,k)$," where:

x and y are the position coordinates at a step;
ψ is the heading measure at a step;
ρ is the reciprocal turn radius employed at the step;
k is the generation level of the incremental step.

The cost function can be: a precomputed database; computed during the trajectory propagation process; or be a hybrid of the two. The position dependent values typically reflect terrain elevation and threat proximity. The ψ parameter can be used in conjunction with position (x,y) to score aircraft aspect angle dependency to threat location or it can simply be used to encourage the aircraft to show a preference for heading in the intended waypoint direction. The turn control, ρ, can be employed to penalize and hence reduce maneuvering.

2.4.1.4 Dynapath Functional Description

A functional block diagram of the Dynapath TF/TA algorithm and Command Generators is shown in Figure 30. The TF/TA algorithm computes a horizontal solution to the trajectory which optimizes the performance measure in the vicinity of the reference ground track. The horizontal solution is handed off to the Vertical Command Generator for an optimization of the TF/TA trajectory over the terrain data associated with the horizontal path. Both solutions result in a computation of consistent commands for the state derivatives.

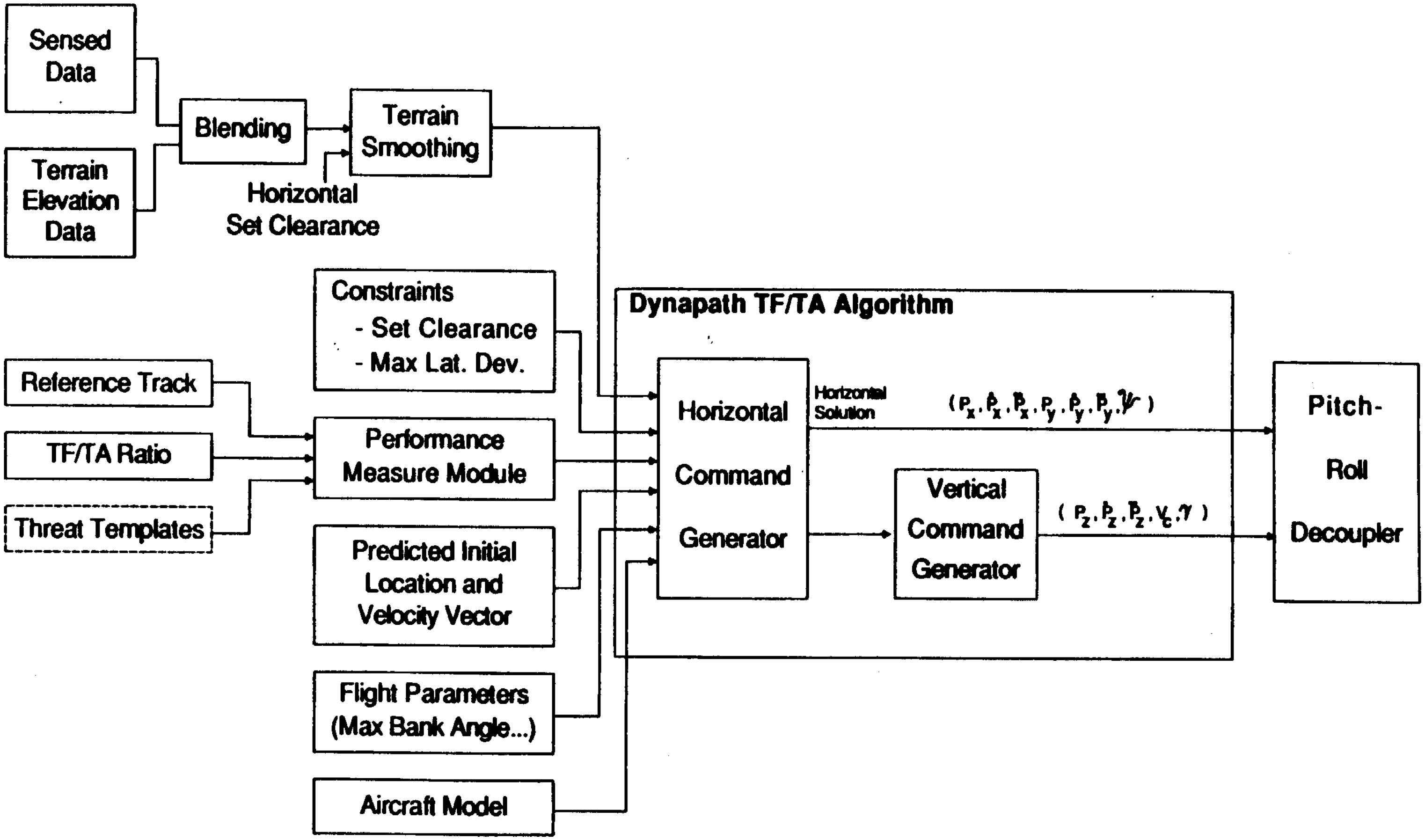

Sensed Data
Terrain Elevation Data
Blending
Terrain Smoothing
Horizontal Set Clearance
Constraints
- Set Clearance
- Max Lat. Dev.
Reference Track
TF/TA Ratio
Threat Templates
Performance Measure Module
Predicted Initial Location and Velocity Vector
Flight Parameters (Max Bank Angle...)
Aircraft Model
Dynapath TF/TA Algorithm
Horizontal Command Generator
Horizontal Solution
$(P_x, \dot{P}_x, \ddot{P}_x, P_y, \dot{P}_y, \ddot{P}_y, \dot{\psi})$
Vertical Command Generator
$(P_z, \dot{P}_z, \ddot{P}_z, V_c, \gamma)$
Pitch-Roll Decoupler

Figure 30. Block Diagram of Dynapath Interfaces

This decoupled approach to trajectory generation wherein the horizontal and vertical paths are separately optimized is a simplification of the overall 3 degree of freedom trajectory optimization. The benefit of this procedure is the reduction in computational complexity. The method assumes the aircraft can simultaneously perform within both horizontal and vertical maneuvering limits.

Inputs and Modules

A major input to the algorithm is the digital terrain elevation data. This data is smoothed in a pre-processing step, which applies safety factors to keep the algorithm from selecting trajectories too close to high frequency peaks in the terrain.

The Horizontal Command Generator (provides a ground track as specified by set of closely spaced ground track points (x_0,y_0), (x_1,y_1), ..., (x_n,y_n). This smoothed command ground track is sent to the Vertical Command Generator, which computes all the vertical command states. The speed may vary as a function of the vertical flight profile, depending on the aircraft model used. For a constant energy model, the speed V_c is not known until after computation of the vertical commands. However, the most suitable implementation for the NASA-NOE simulation environment has been to assume a constant velocity system. Within limits, the capability for constant velocity is assured by limiting the climb/descent profile in the algorithm compared to the true helicopter maneuverability as discussed in Section 2.2.1.2.

The Vertical Command Generator module receives the terrain profile associated with the horizontal path solution and optimizes for the the TF/TA/NOE trajectory which most closely follows the terrain subject to the set clearance altitude constraint and the aircraft maneuverability limits. The profile is illustrated in Figure 36.

The inertially referenced commands are then passed through to the pitch-roll decoupler. This provides an interface to the flight control system and serves a tracking function of guaranteeing adequate authority to the vertical channel to maintain altitude, while assuring that lateral deviations from the commanded trajectory are minimized.

2.4.1.5 The Dynapath TF/TA/NOE Algorithm

The Dynapath algorithm is a mixture of Dynamic Programming (DP) and tree searching. The tree structure has been implemented in a way which minimizes the amount of computation associated with the kinematics of the aircraft and the Dynamic Programming to selectively reduce the number of possible trajectories. So, basically, the problem is solved by a simple forward running Dynamic Programming algorithm where the state transitions are handled by a tree structure.

As a preview, the advantages of this hybrid approach over the conventional Dynamic Programming approaches are:

- Aircraft kinematics are incorporated explicitly. The everywhere smooth trajectory, in having no "kinks," does not require any smoothing that displaces the trajectory laterally. This is extremely important due to horizontal set clearance considerations.

- Aircraft controls in inertial coordinates are computed directly from the trajectory parameters associated with the aircraft model.

- Coordinate transformations associated with either the terrain data orientation or the solution axis orientation are not needed. This is very important since Dynamic Programming usually involves selection of an x-y coordinate frame which, once selected, tends to impose a particular direction of travel. To circumvent this problem, coordinate transformations and data interpolation are often required. Such problems are avoided in this development.

- Heuristics can be added to reduce computation time.

In the following, the "decoupled" approach will be discussed in the generation of the TF/TA trajectory. This approach finds the lateral ground track first, followed by a determination of the vertical commands.

2.4.1.5.1 Decoupled Procedure

In the decoupled approach a ground track is found by essentially assuming that the aircraft can fly perfectly in the set clearance surface. This surface is a surface above the smoothed terrain surface but displaced by a constant set clearance bias. The TF/TA tradeoff is made under this assumption, resulting in the lateral ground track. The vertical command generator then relaxes the assumption that the aircraft flies perfectly at the set clearance altitude, and treats the set clearance altitude as a minimum altitude constraint.

2.4.1.5.2 Ground Track Computation

In computing the set of potential maneuvers of the aircraft, we start with a consideration of coordinated turns. The two dimensional trajectory of the aircraft is a function of speed and bank angle. A change in the bank-angle in turn affects the reciprocal instantaneous radius of curvature ρ.

A time scale quantization of one second is a suitable unit for the framework of assigning maneuvers since an aircraft typically requires 1-2 seconds to roll from one banked turn maneuver to another.

In the development of Dynapath, five bank angles were selected to represent the aircraft at discrete lateral maneuvering capabilities within its performance limits.

A five state tree and a corresponding discretization in time of 1 second, which is approximately how long it takes to go from one state to a neighboring state were also found to be suitable in terms of finding solutions in a real-time computing environment. Note that a finer quantization in time would cause the tree of possible trajectories to increase--with corresponding computational increases--while coarser quantization in time will be seen to undersample the performance measure, the latter being associated with the terrain data.

The reciprocal radius of the turn radius $\rho=1/r$ was selected as the control variable. Note that this measure doesn't vanish with zero bank angles and can be expressed relative to gravity (g), velocity (V) and bank angle (ϕ) as:

$$\rho = \frac{g \tan \phi}{V^2}$$

A tree is generated using ρ as the control variable. Actually, all possible discrete values of ρ are used initially to exhaustively generate every branch of the tree for the first N seconds. Each node of the tree corresponds to a time increment (typically 1 second) from the previous node. At each node k of the tree the following information is stored:

$$S_k = \begin{cases} \text{Position } (x,y) \\ \text{Heading } \psi \\ \text{Parent: node that has generated node } k \\ \text{Cost: cumulative cost up to and including the present node (for} \\ \qquad \text{the performance measure being used)} \\ \text{Curvature control used to arrive at node } k \end{cases}$$

Every time a new node k is generated, S_k is computed using $S_{parent(k)}$ and a transition matrix to be defined below.

The curvature controls correspond to the bank angles selected for the maneuvering of the aircraft. They are typically quantized in five values corresponding to the maximum bank angle in each direction, half the maximum bank angle, and straight flight. The five discrete ρ's are:

$$\rho = 0,\ \pm \frac{g}{v^2} \tan(\phi_{max}/2),\ \pm \frac{g}{v^2} \tan(\phi_{max})$$

The corresponding controls are referred to as: 0, ±1, ±2 where negative controls direct a right turn and ±2 directs use of the maximum permissible bank angle.

Because of limitations on the roll-acceleration of the aircraft, ρ is limited as to how much it can change at each transition. Accordingly, ρ can only change by one control measure at each time interval. also, the ±1 controls are often used as transition states requiring the next control selection to continue descending/ascending as dictated by the previous command.

An Example Tree

Starting with $\rho = 0$ and arbitrary heading ψ, an example tree is given in Figure 31.

This is a tree of n=3 stages, or time steps. Of course, the branch length in Figure 31 has been exaggerated to better demonstrate the tree structure.

Note the way nodes are numbered. This numbering scheme is the standard used throughout the algorithm, mainly because the nodes at the last branches --called "end nodes"--then have sequential numbers. This scheme simplifies the process of allocating nodes in computer memory and aids in locating and propagating branches.

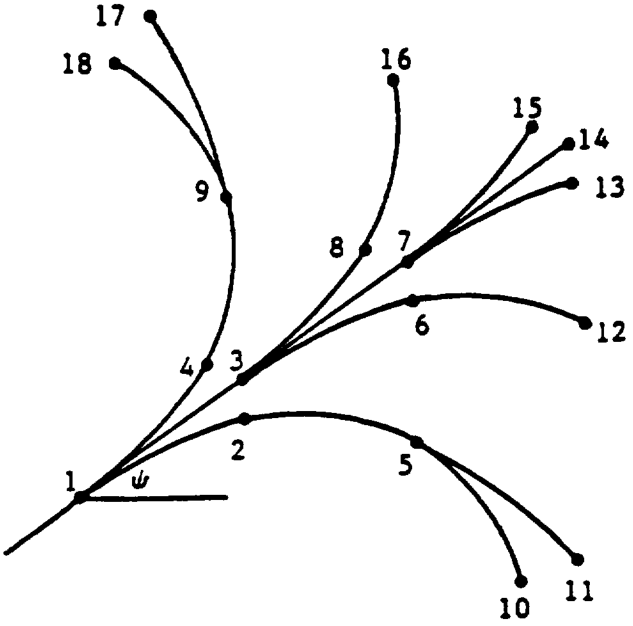
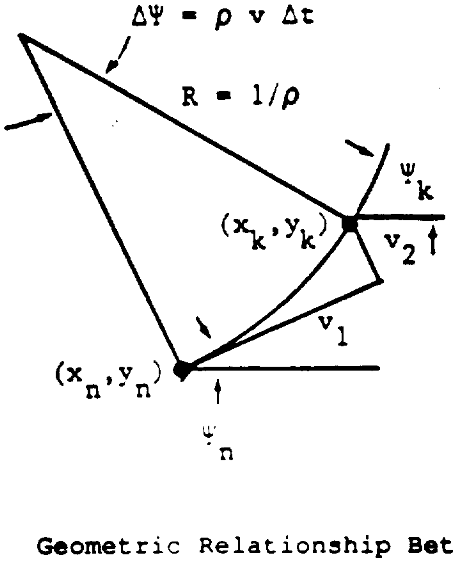


Figure 31. An Example Tree



$$v_1 = \frac{1}{\rho}\sin\Delta\Psi$$

$$v_2 = \frac{1}{\rho}(1-\cos\Delta\Psi)$$

Figure 32. Geometric Relationship Between Parent and Offspring

Transition Matrix

It is efficient to develop an iterative formulation in which the elements of the state vector S_k, the k'th node, are related to the parent node to k, designated n = parent (k). Using such a iterative formulation, it is then straight forward to trace through the nodal structure to retrieve any path that satisfies a criterion such as lowest cost.

Consider the geometry shown in Figure 32. From parent location (x_n, y_n) with heading ψ, the offspring at k is simply a rotation about the z axis through angle $\Delta\psi$ where $\Delta\psi = \rho v \Delta t$. The (x, y, ψ) elements of the state vector S_k are thus given by:

$$\begin{bmatrix} x_k \\ y_k \\ \psi_k \end{bmatrix} = \begin{bmatrix} \cos\psi_n & -\sin\psi_n & 0 \\ \sin\psi_n & \cos\psi_n & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \vec{v}_\rho + \begin{bmatrix} x_n \\ y_n \\ \psi_n \end{bmatrix} \qquad (1)$$

$$\vec{v}_\rho = \begin{bmatrix} \frac{1}{\rho} \sin(\Delta\psi) \\ \frac{1}{\rho} (1 - \cos(\Delta\psi)) \\ \rho v \Delta t \end{bmatrix}$$

Here $\vec{v}_\rho$ is a constant vector associated with turns to the left, consistent with Figure 32. In particular it applies to discrete quantizations of ρ. Symmetrically opposite right turns are obtained by appropriate sign changes. Wings level is formed by taking the limit $\rho \to 0$ in the expression for $\vec{v}_\rho$.

Notice what quantization in ρ has accomplished. A computationally tractable form for the state vector is obtained in Equation 1. Also, note that the relatively large changes in bank angle are related to the one quantity that remains relatively slowly varying--the curvature ρ. Using this parameterization, a sequence of state vectors always traces out a smooth trajectory. There are none of the usual quantization artifacts that usually haunt Dynamic Programming approaches to trajectory generation.

Furthermore, by precomputing and storing the sin and cos values for the transitions as well as the discrete turn increments, the computational load is significantly reduced.

The other state variables in S_k are given by:

parent (k) = n by definition

cost (k) = cost (n) + cst (k) where cst (k) is the incremental cost of being in the node k according to the definition of the cost functional. As already noted, cst is the "valley seeking" cost functional. In general it can be replaced by any function cst (x,y,ψ,ρ,k) as discussed in Section 2.3.1.3).

ρ_k = the curvature transition as discussed already; the curvature here is actually the control variable with the value used to arrive at k stored as the last element of the state vector S_k.

The above state description can be propagated forward in time for all possible controls--the curvature quantizations--to generate all nodes of the tree. By looking across the branches at the very end, the end nodes, the optimal cost can be selected and the optimal trajectory would thus be known. However, the data storage and computational requirements increase exponentially with the number of generations.

Constraint Pruning Within the Tree

Given an initial position, heading and curvature, we construct a complete tree representing all acceptable paths that the aircraft can follow for the next N seconds. Note that N is the level number of the tree (the depth) because each node transition represents 1 second. Note also that construction is accomplished starting from the aircraft's current heading regardless of the inertial axes. The nodes, however, have (x,y) positions which use the same inertial reference used for the associated terrain data. Thus, no rotations are needed apart from the state vector propagation in Equation 1.

At tree generation time, branches can be discarded according to several possible criteria prior to a cumulative cost comparison. The use of such criteria is denoted as constraint pruning. The specific criteria used in the present approach are:

a. A node under consideration must not exceed the maximum lateral deviation from the reference path.

b. The heading at the end node of a tree must lie within a user-specified angle range measured from the reference path direction.

c. The end node of a tree must exhibit net forward progress along the reference path with respect to the starting node of the tree.

Dynamic Programming Overlay

So far the presentation may have given the impression that there is only one single tree. In fact, the algorithm might be thought of as growing many trees, selectively pruning them, then growing more, etc., until there is virtually a uniformly dense forest of only the best trees. From this forest the single best tree corresponding to the optimum in the performance measure is selected. It is the Dynamic Programming overlay that accomplishes this selective pruning. As will be seen, it compares branches arriving in the same cell on a cost basis, following the constraint pruning described above.

The end nodes of the initial and later trees are classified into a Dynamic Programming "overlay," as shown in Figure 33. This is shown as a rectangular grid that is oriented along the reference track. (Other grid geometries have also been used during the development; the particular shape of the grid can be altered if desired.) Subdivisions are indicated as a three-dimensional spatial classification of the space according to the zone, division, and heading dimensions. The heading subdivisions are divided according to an angular classification into one of several possible cells (possible azimuth directions). Thus, the end nodes of a tree are classified according to a cell of dimensions Δx, Δy, and Δψ.

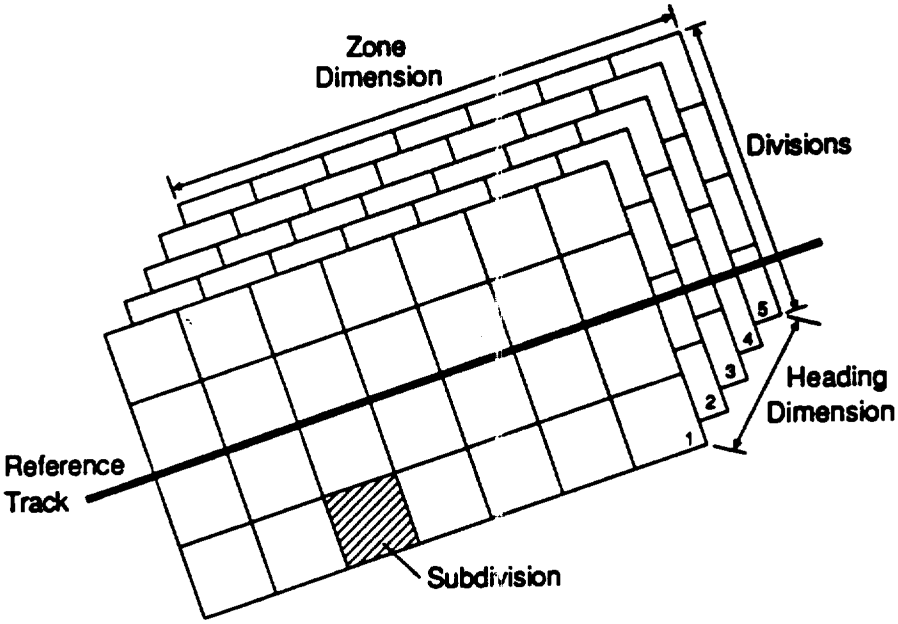


Figure 33. Dynamic Programming Overlay

The number of zones, divisions, and heading subdivisions are selected to correspond to the degree of pruning which is desired. The coarser the resolution of the overlay, the more aggressive will be the pruning process. With fewer subdivisions, fewer candidate paths survive the pruning process to start new generations of trees. Conversely, as the number of partitions is increased (smaller increments in zones, divisions, and/or headings) fewer candidate trajectories will be compared in each cell in the pruning process. As a result, more paths are retained to propagate new generations of tress. A greater number of potential paths can therefore be compared in selecting the overall optimum trajectory.

The increase in number of paths generated, however, proportionally increases the degree of computation required in generating a solution. When the

Dynapath algorithm is used as a real-time component for path optimization, the cell size must be balanced to the computer processing rate. Typical subdivisions currently used on VAX minicomputers use about 20 zones, 20 divisions, and 5 heading subdivisions.

The zone and division components of the dynamic programming overlay are independent of the terrain data grid (x,y values) which are used in the calculation of the trajectory and its nodes.

As already indicated, there are many trees that are grown rather than one single large tree. For a given tree, a DP state for an end node "k" contains a label designating the trunk (source) of the end node, the cumulative cost to that end node, as well as state and control information.

Note that many end nodes may have the same source, namely, the end nodes for a given tree. Also, Dynamic Programming states will be selected on the basis of the best cumulative cost at the end nodes, but do not require storage of the full set of controls and states in traversing from the tree source to a given end node. In short, the DP states "leapfrog" from end nodes to trunks without storing the intervening branches. However, note that retrieval of the immediately preceding curvature control is all that is necessary to restart generation of a new tree according to the transition scheme of Equation 1.

Optimization Procedure

Starting from the initial position and heading in the patch, an initial N stage tree is generated. The value of N is typically three to five, i.e., 3 to 5 seconds of flight time. The initial tree corresponds to approximately 9 to 27 end nodes (if the previous control was 0) and 18 to 60 total nodes including the initial trunk node. (For an illustration of this process, refer to Figure 39.) Pruning of this tree and subsequent trees will occur according to criteria such as the maximum lateral deviation from the reference track being exceeded. After pruning, new trees are generated from the remaining end nodes. These new trees are pruned in turn, and the process continues.

In parallel with the pruning, the Dynamic Programming selection proceeds. As the tree is generated, the cell corresponding to each end node is computed. If the cell is empty, the end node, including its cost, is registered as being in the cell. If the cell is already occupied by an end node, the cost of the current end node is compared with the previously registered cost and the end node with lower cost is kept. This forms the basis for the Dynamic Programming (DP) operation for selecting the best trees.

Many trees are used by this technique in propagating to the end of the patch. Once the end nodes of the last trees are past the last zone in the patch shown in Figure 33, the optimal patch is determined by selecting the end node with the lowest cumulative cost. (Additionally, various end node boundary conditions such as a maximum lateral deviation or heading with respect to the reference track can be imposed.)

The optimum path is retrieved by tracing backward through the DP structure until arriving at the initial tree. This is possible because we have kept track of the source at every stage. We note that the full set of controls--in one second quantizations--is available for the each tree due to the way the solution is constructed and stored. The optimal solution is of course based on the uniform 1 second quantization over the entire patch length due to the manner in which the DP solution is constructed.

A portrayal of this overall ground track optimization process is shown in Figure 34.

Features of Dynapath TF/TA Algorithm

Several features of this algorithm are important. First, it is different from conventional DP in that all necessary information is stored to smoothly generate a new tree, in addition to the storing of the DP state. This necessary information is, of course, the initial set of conditions S = (parent, cost, x,y,ψ, ρ) for the next tree. It is important to note that the end node may lie anywhere within the cell. The classification of an end node within a cell does not force any quantization of the DP state to the overlay structure.

Second, the use of an angular quantization ψ for each cell is significant. Not only are the costs compared within a spatial quantization--the subdivisions--they are compared only if they lie with the same heading quantization within the subdivision. Neglecting this consideration can otherwise lead to a significant loss of optimality by pruning away slightly more costly paths which are headed to more promising/less costly terrain.

Third, note that even though there is a clear forward progress direction defined (in the direction of increasing zone number), a state does not always propagate to the next zone. For this reason, every cell in increasing zone number is scanned to insure that all the states are propagated. From each occupied cell a new tree is generated, which in turn is classified according to its end node cell. For proper operation, end-points that have zone numbers lower than or equal to their sources are discarded. This, of course, is compatible with the forward progress assumption. The nodes that fall beyond the last zone are candidates for the optimal path, and the one that has the lowest cumulative cost is kept as the "winner."

Finally, note that the individual end nodes generally arrive at a given cell at different times, depending on their path history. The processing accounts for this by putting no constraints on time of arrival but only on method of arrival. This is to say that nodes entering a cell may have taken vastly different routes with the most meandering taking the most time. The "stragglers" are always allowed to catch up before the processing continues.

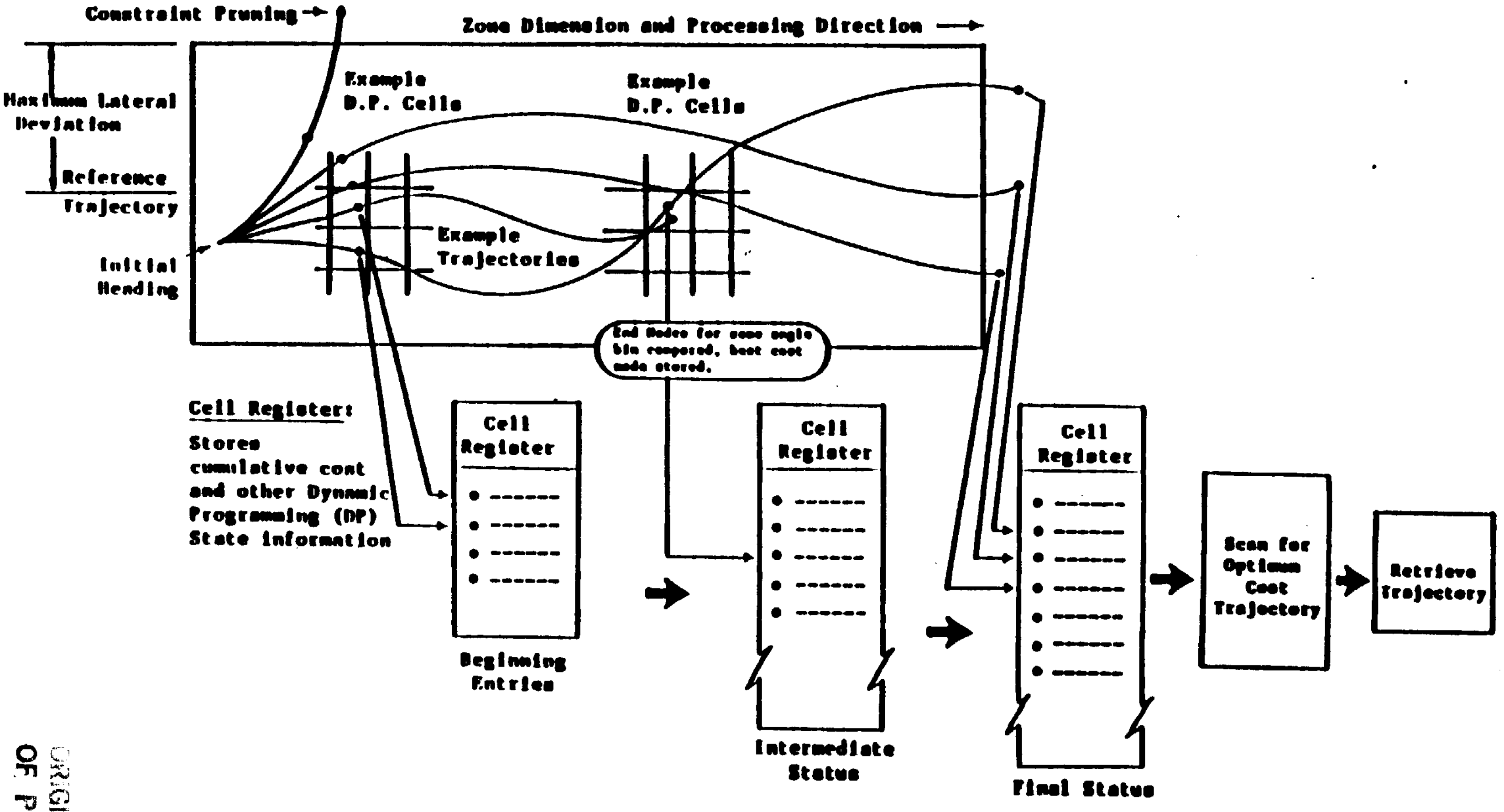

Constraint Pruning
Zone Dimension and Processing Direction
Maximum Lateral Deviation
Reference Trajectory
Initial Heading
Example D.P. Cells
Example D.P. Cells
Example Trajectories
End Nodes for same angle bin compared, best cost node stored.
Cell Register:
Stores cumulative cost and other Dynamic Programming (DP) State information
Cell Register
Beginning Entries
Cell Register
Intermediate Status
Cell Register
Final Status
Scan for Optimum Cost Trajectory
Retrieve Trajectory

Figure 34. Dynapath Processing Steps Within Patch

2.4.1.5.3 Vertical Trajectory

Prior to generation of the vertical commands, the vertical trajectory must be generated. To achieve this, a dynamic programming procedure similar to the horizontal path generator is used. In this case, a terrain following set of deviations are constructed where the terrain values are known by extracting the terrain elevation associated with the (x,y) values of the digital map at each step of the horizontal path. The heading deviations are replaced with flight path increments which are assigned within the climb and descent angle parameters assigned to the aircraft model. The approach will be described in the following.

Controls

As in the horizontal case, the controls will be taken to be path curvature, this time in the vertical plane. Four curvature quantizations are selected corresponding to 2 positive incremental normal g's, zero incremental normal g, and negative incremental normal g. The curvature control designated ρ_v, where

where $\rho_v = \dfrac{N_z - g}{v^2}$, and N_z is the incremental normal g load.

The aircraft model in Section 2.4.1.6 will be used to relate these curvatures to the normal acceleration seen by the aircraft. In this decoupled approach, the vertical curvature controls are selected independently of the horizontal channel. Representative incremental normal g loads vary between (-1 to +2g) for a tactical aircraft in a severe environment to (-.25 to +.25) for a helicopter in a near NOE environment.

The quantizations in ρ_v are chosen, using the aircraft model, to correspond to the normal load factor limits. Even though four quantizations are used, more quantization levels could be used. This, together with limitations on transitions, can be used to account for pitch jerk constraints. (Pitch jerk constraints have not been included within this development.)

The Vertical Tree Structure

The states S_k of node k is defined by

$$S_K = \begin{cases} s & \text{cumulative distance along ground track (x,y)} \\ z & \text{altitude} \\ \gamma & \text{flight path angle} \\ \text{parent} & \text{Node that generated this node} \\ \text{cost} & \text{cumulative cost up to and including this node} \\ \rho_v & \text{curvature control} \end{cases}$$

This state vector is completely analogous to that used in the ground track development. The cost can have any functional form that tends to "push down" the trajectory to the set clearance altitude. We have used a cost functional

$$cst(s) = H^2(s),$$

consistent with the functional used for the TF/TA tradeoff. Note that s is the cumulative distance measured along the ground track of the horizontal path and that (x,y) = f(s) and is known from the ground track computation; thus H(s) = H(x,y).

State Transitions

The state transitions are shown in Figure 35.

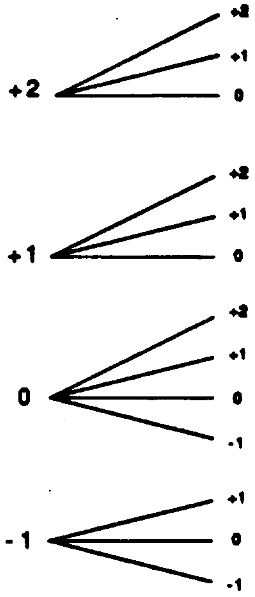


Figure 35. State Transitions for Vertical Trajectory

Transition Matrix and Node Generation

The transition matrix and node generation for vertical trajectory generation are analogous to the horizontal case. From state z_n, with flight path γ, the offspring at k is simply a rotation about the pitch axis through the angle $\Delta\gamma$ where $\Delta\gamma = \rho_v V \Delta t$. The (x,γ) elements of the state vector S_k are thus given by:

$$\begin{bmatrix} z_k \\ \gamma_k \end{bmatrix} = \begin{bmatrix} \sin\gamma_n & \cos\gamma_n & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \vec{v}_\rho + \begin{bmatrix} z_n \\ \gamma_n \end{bmatrix}$$

$$\vec{v}_\rho = \begin{bmatrix} \frac{1}{\rho_v} \sin(\Delta\gamma) \\ \frac{1}{\rho_v} (1 - \cos(\Delta\gamma)) \\ \rho_v v \Delta t \end{bmatrix}$$

Here $\vec{v}_\rho$ is a constant vector associated with pitch maneuvers, consistent with Figure 35. In particular it applies to discrete quantizations of ρ.

As in the horizontal path generation process, by precomputing and storing the sin and cos values for the transitions as well as the discrete flight path angle increments, the computational load is reduced.

Optimization and Pruning

Extensive pruning can be done within the tree because of strict limitations on climb and dive angles. Also, a node is pruned when ever it lies below the set clearance altitude. (Thus, the set clearance altitude is treated as a hard constraint. This constraint can be softened if desired for minor excursions below the set clearance altitude.)

In carrying out the pruning, higher solution resolution is obtained by checking the set clearance altitude constraint at each mid-branch in addition to the check at each node.

After propagation to the end of the patch, the best solution is then selected. The process is sketched in Figure 36.

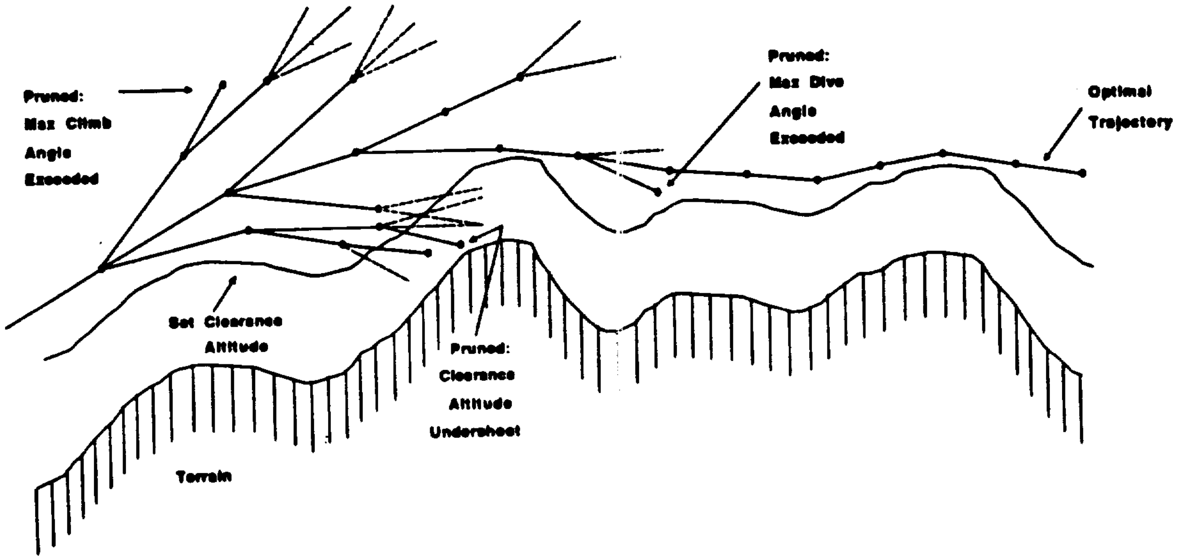


Figure 36. Vertical Solution Procedure

Once the vertical solution is known, then the entire three dimensional trajectory is known in terms of a sequence of positions (x,y,z) together with the sequence of associated horizontal and vertical curvatures ρ and ρ_v respectively. From this parameterization, the inertial accelerations can be calculated.

2.4.1.6 Aircraft Model

The aircraft model applied to the NOE Dynapath algorithms is shown below. The equations will be summarized here and related to the controls used in Sections 2.4.1.5.2 and 2.4.1.5.3.

With reference to Figure 37, the accelerations perpendicular and parallel to the velocity vector are given by:

$$\begin{aligned} v\,\dot{\psi}\cos\gamma &= n_z \sin\phi \\ v\,\dot{\gamma} &= n_z\cos\phi - g\cos\gamma \\ \dot{v} &= -g\sin\gamma \end{aligned} \tag{3}$$

The Equation for $\dot{v}$ is a constant energy equation. However, in the current implementation of Dynapath v is held constant, corresponding to a varying energy. This implementation is consistent with the initial NASA application to evaluate Dynapath in a real-time mode supporting constant velocity NOE flight.

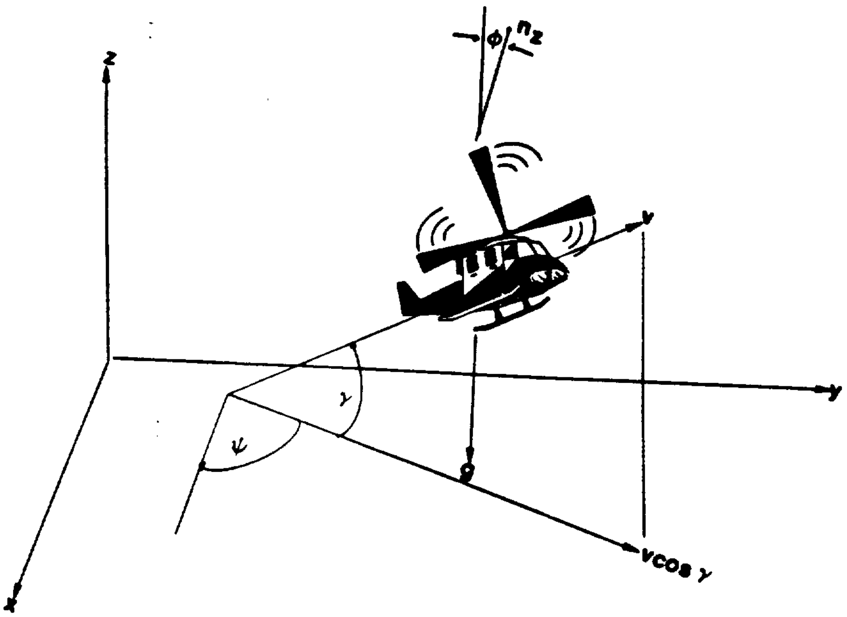


Figure 37. Aircraft Model Variables

The $\dot{\psi}$ and $\dot{\gamma}$ terms, being rotations in the horizontal and vertical planes, can be expressed in terms of rotations using the reciprocal radius controls introduced in the previous subsections. For clarity, the ρ term used in the horizontal path generation will henceforth be designated ρ_H. The rotations are:

$$\begin{aligned} \dot{\gamma} &= \rho_V \; v \\ \dot{\psi} &= \rho_H \; v \cos \gamma \end{aligned} \tag{4}$$

In both of these equations ρ_V and ρ_H may be negative, corresponding to "negative g's" in the vertical plane and to right turns in the case of the convention defining ψ in Figure 34.

Meanwhile the inertial axis motion, at constant velocity, is given by:

$$\begin{aligned} \dot{p}_x &= v \cos \gamma \cos \psi \\ \ddot{p}_x &= -n_z \; (\sin \psi \sin \phi + \cos \psi \cos \phi \sin \gamma) \\ \dot{p}_y &= v \cos \gamma \sin \psi \\ \ddot{p}_y &= n_z \; (\cos \psi \sin \phi - \sin \psi \cos \phi \sin \gamma) \\ \dot{p}_z &= v \sin \gamma \\ \ddot{p}_z &= n_z \cos \phi \cos \gamma - g \end{aligned} \tag{5}$$

The position information (p_x, p_y, p_z) in one second intervals is of course known from the the trajectories determined in Sections 2.3.1-4 and 5, and the heading ψ and flight path angle γ for each position are known as well from that same development.

Meanwhile, the acceleration terms can be expressed in terms of the ρ_H, ρ_V controls by inserting Equation (4) into Equation (3) and then using the results in the appropriate acceleration expressions Equation (5).

The final form is:

$$\begin{aligned} \dot{px} &= v \cos \gamma \cos \psi \\ \ddot{px} &= -\rho_H \; (v \cos \gamma)^2 \sin \psi - [\rho_V \; v^2 + g \cos \gamma] \cos \psi \sin \gamma \\ \dot{py} &= v \cos \gamma \sin \psi \\ \ddot{py} &= \rho_H \; (v \cos \gamma)^2 \cos \psi - [\rho_V \; v^2 + g \cos \gamma] \sin \psi \sin \gamma \\ \dot{pz} &= v \sin \gamma \\ \ddot{pz} &= \rho_V \; v^2 \cos \gamma + g \; [\cos^2 \gamma - 1] \end{aligned} \tag{6}$$

We note again that the controls ρ_H and ρ_V have both positive and negative values.

Note that the bank angle ϕ has been eliminated from the expression for p_x in Equation (6). In terms of the controls, its value is:

$$\phi = \tan^{-1} \left(\frac{\rho_H \, v^2 \cos^2 \gamma}{\rho_V \, v^2 + g \cos \gamma} \right) \tag{7}$$

Thus, all inertial terms as well as bank angle ϕ can be expressed in terms of the trajectories and associated controls ρ_H, ρ_V at the one second quantization scale.

2.4.1.7 Dynapath Algorithm Summary

The converted Dynapath algorithm process flow for a single patch computation is shown in Figure 38. The 2-D horizontal path generating algorithm first determines the optimum ground track and then employs similar techniques in a separate vertical path generating module.

Table 12 summarizes the key parameters used in the Dynapath algorithm. The digital terrain data and the cost function performance measure are pertinent to use of the map and are chosen by the user.

The horizontal and vertical set clearance values are optimization parameters, as are the user selected TF/TA ratio and the maximum lateral deviation which the algorithm is allowed in searching for the best horizontal trajectory.

The essential flight parameters which affect the Dynapath algorithm are the acceleration relevant terms such as normal load, max bank angle, and roll acceleration, and also the velocity and flight path angle limits.

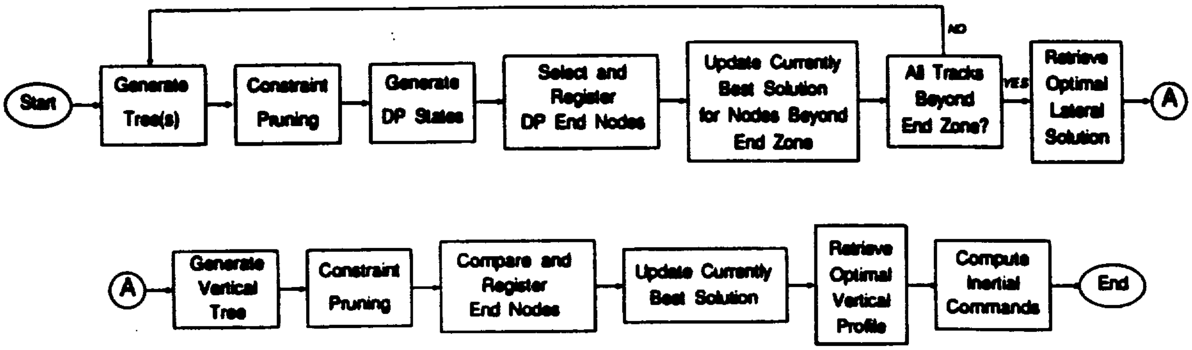

NO
Start
Generate Tree(s)
Constraint Pruning
Generate DP States
Select and Register DP End Nodes
Update Currently Best Solution for Nodes Beyond End Zone
All Tracks Beyond End Zone?
YES
Retrieve Optimal Lateral Solution
A
A
Generate Vertical Tree
Constraint Pruning
Compare and Register End Nodes
Update Currently Best Solution
Retrieve Optimal Vertical Profile
Compute Inertial Commands
End

Figure 38. Dynapath Process Flow

Table 12. DYNAPATH TF/TA ALGORITHM SUMMARY TABLE

INPUTS	COMMENTS
Blended, Smoothed Terrain Elevation Data	Data formatted to standards of Level 1 DMA Terrain Data, blended with real time sensor data, and smoothed for horizontal set clearance constraint
Performance Measure	(Valley seeking) $\sum_{i=1}^{n} (H_i^2 + \omega D_i^2)$
OPTIMIZATION PARAMETERS	
Horizontal Set Clearance	Accounted for by off-line smoothing of Terrain Data on scale set by $(H_{set})_{hor}$ value
Vertical Set Clearance	Constant bias AGL measured from smoothed terrain, supplied as user input
Maximum Lateral Deviation from Reference Path	User input
TF/TA Ratio - ω	User input
Initial Boundary Conditions	Supplied within simulation from previous patch
FLIGHT PARAMETERS	
Normal Load Factor n_z	Used to set $\rho_V = \frac{n_z - g}{v^2}$ for $\phi, \gamma = 0$ in Eq. 3
Max Bank Angle ϕ_{MAX}	Used to quantize $\rho_H = \frac{g}{v^2} \tan\phi$ from Eq. 7 with $\rho_V, \gamma = 0$
Airspeed	From aircraft model and simulation
Flight Path Angle	Explicitly used in vertical plane pruning, limit angles are user inputs
Roll Acceleration	Used to establish time quantization and transition scheme
Aircraft Model	Eq. 3 with $\dot{v} = 0$

2.4.1.8 Conversion of Dynapath Code for NASA Application

2.4.1.8.1 Introduction

As discussed in the beginning of this section, a significant level of effort took place in adapting the original Dynapath code to the NASA Helicopter NOE flight regime. The conversion effort was driven principally by the more restrictive aircraft maneuver limits, the goal of increasing ride quality, and the need to maintain real-time code execution on a slower computer than was available for the prototype tests.

Table 13 lists the key efforts made to enhance Dynapath under the contract. Most efforts had a beneficial effect on advancing the technology of the software, while one effort may hold promise in future applications, but was abandoned in the quest for sufficient speed of execution.

Table 13. Specific Modifications Made for Helicopter NOE Environment

		Effect on Algorithm
• Increased frequency of pruning in lateral and vertical path generation	-	increased speed
• Increased number of vertical controls	-	smoother ride
• Filter terrain profile before vertical path generation	-	increased speed
• Accommodate vertical trees extending below clearance altitude by inflicting a high-cost penalty (see Reference Diagram on next page)	-	program stability
- attributes to vertical velocity component in TF	-	increased model fidelity and speed
- Enhanced by precomputing terrain tables		
• Evaluate epsilon controls for a smooth lateral path	-	smooth trajectory at increased run time

2.4.1.8.2 Lateral Path Code Developments

The original Dynapath model was configured to generate a single tree for the first 10 seconds of flight, generating path sections of 1 second intervals. After pruning those trees not already clipped by constraints during the first phase of path generation, the time step was increased to 2 second intervals and a revised control set was used to propagate trees in 3 stage increments out to a 30 second path length. The technique produced an enormous number of candidate trajectories and then selected the least cost, or optimum, from the set. The execution speed via this technique proved extremely slow on the VAX system.

The lateral path generation code was generalized to allow for a reduced number of stages for the initial tree and a variable size for subsequent trees. A performance study was conducted to search for a balance between speed of execution and path optimization. For the terrain data bases in use, as few stages as 3 were found to solve for the optimum trajectory reasonably well when the secondary trees were also generated at one second intervals. Figure 38 (left side) diagrams the lateral path generation control structure initially used and itemizes the number of nodes created within this procedure.

2.4.1.8.3 Development of Epsilon Controls

The standard set of five controls used in the Dynapath algorithm were intended to provide a balance between using as few controls as possible to maximize speed of computation in a real-time process, yet use the controls in such a way that the maximum maneuvering performance of the aircraft could be employed. Maximum maneuverability is only occasionally necessary to find and exploit the best terrain features for TF/TA flight.

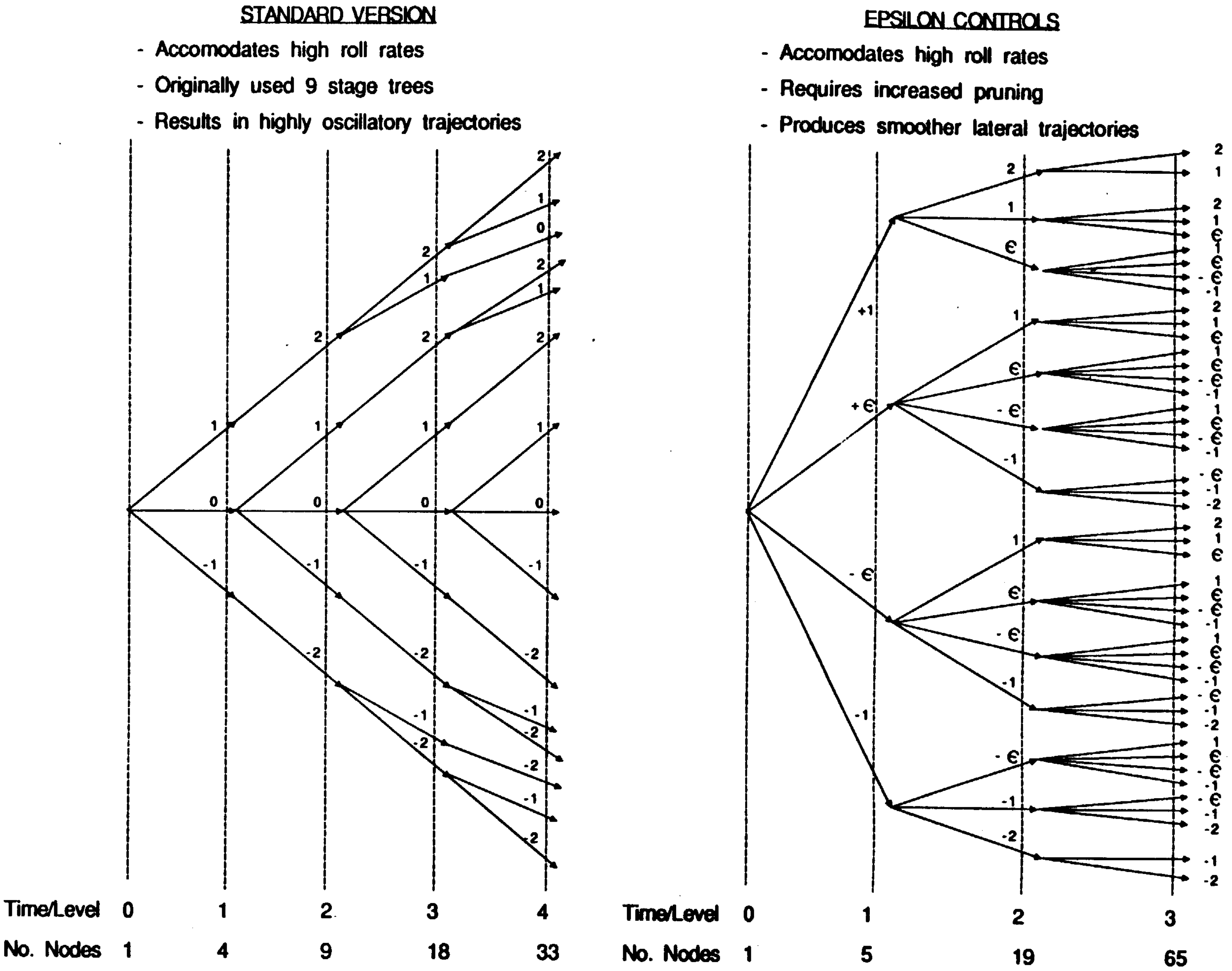
As a result, during the majority of the flight path generation, when only minor differences in terrain are detected, rather aggressive maneuvers are still used to produce the lateral trajectory. The oscillatory paths produced by this process are both uncomfortable over extended periods in a manned aircraft and call for a large amount of activity in controlling the aircraft, an undesirable feature in both manned or automatic flight modes.

The undesirable oscillations tend to intensify with the use of high TF/TA ratios because the cost penalty associated with being even slightly off the center line of the route increases rapidly. In addition, since any use of bank angle control requires the aircraft to continue to the maximum bank angle before returning to wings-level, slight deviations from the nominal heading requires maximum performance corrections.

An attempt which was made to solve this problem was to introduce two more controls to the search process. These controls were added about the zero heading control. The zero heading control was removed, resulting in a net increase of only one control. However, the interaction between controls was revised to allow half bank controls to regenerate or to return to either epsilon control as well as to proceed to max bank angle.

Figure 39 illustrates the epsilon control relationship and enumerates the number of nodes created at each step. Note that the number after thee stages is larger than the baseline Dynapath by greater than a factor of three. The computational explosion diminished the speed of execution accordingly as is illustrated in Figure 40, but it also produced very smooth trajectories.

In summary, the lateral path modifications to Dynapath served to speed the code execution and to smooth the trajectory, but additional modifications were necessary at the conclusion of the effort under the funding phase of this contract.



STANDARD VERSION

- Accomodates high roll rates
- Originally used 9 stage trees
- Results in highly oscillatory trajectories

EPSILON CONTROLS

- Accomodates high roll rates
- Requires increased pruning
- Produces smoother lateral trajectories

Figure 39. Lateral Path Generation Techniques

2.4.1.8.4 Vertical Path Code Development

The original Dynapath vertical path generation software extracted the terrain profile of the lateral path and solved for the lowest vertical trajectory by generating trees with a set of three controls: max pull-up, zero incremental g, and max nose over. The climb and dive angle limits of the aircraft and the terrain clearance attitude served as constraints to prune the nodes during tree generation.

The process served to prove the concept of generating a flyable vertical path, but resulted in an extremely rough ride. Furthermore, a significant portion of the overall code execution time was spent in the vertical path generation module. The initial modifications to the vertical path generation module focused on reducing the number of nodes generated between pruning and in adding an intermediate positive vertical control (50% of max acceleration). The additional control resulted in a significant increase in "smoothness of ride" and remains in use today. However, the computation time required for vertical path generation, though reduced, was still significantly large due to the large number of paths which were generated. Upon inspection, most of the propagated paths resulted in altitudes far above the terrain clearance altitude and had cost values greatly in excess of the optimum path. It was observed that other propagated paths which were constraint pruned by the set clearance altitude, could have been pruned several generations earlier if it were recognized that the downrange terrain gradient exceeded the maximum angle of climb.

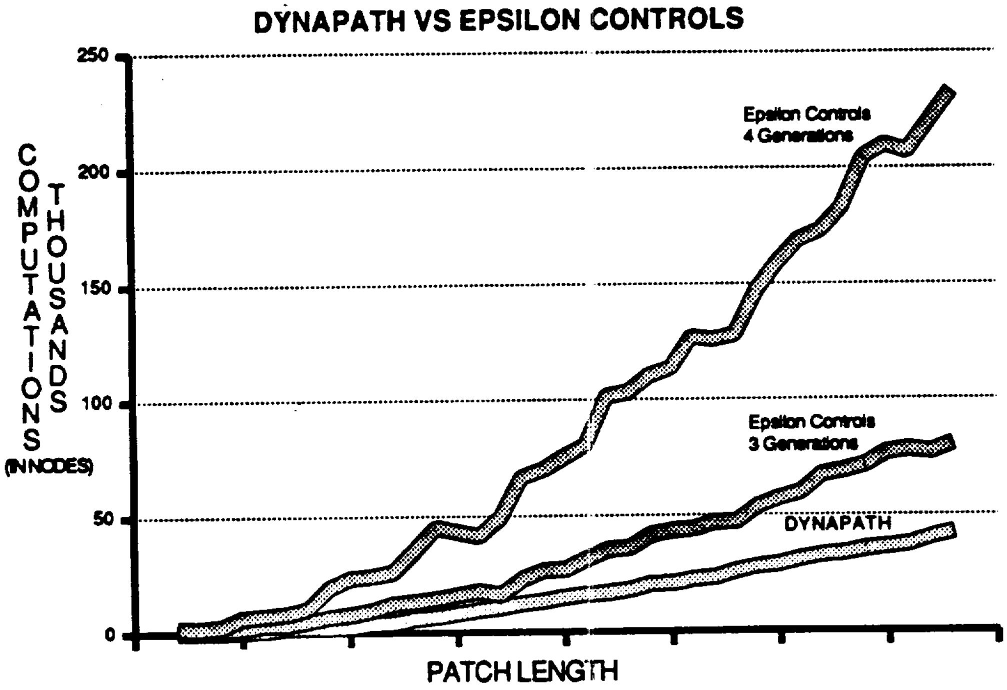


Figure 40. Number of Computations

A terrain filter algorithm was introduced which creates a smoothed set clearance profile to reflect the climb and dive limits of the aircraft. Vertical altitude zones were assigned parallel to the smoothed set clearance and unevenly spaced as illustrated in Figure 41 to aggressively prune vertical paths which were removed from the minimum allowable altitude. The introduction of these measures greatly reduced the number of paths without any loss of fidelity in deriving the optimum vertical path.

The vertical path performance measure was revised to accommodate flying below the smoothed set clearance altitude. Previous to this, simulations would crash if the aircraft dipped even one foot below the constraint. A slight penetration of this constraint across a peak might not be a serious breach of the desire to fly safely as low as possible. The performance measure heavily penalizes flying below the set clearance, but was found to accept, as optimum, the brief excursions characteristic of skimming rough terrain.

2.4.1.8.5 Other Modifications

Another problem which had been occurring involved a foreshortening effect related to the correspondence of lateral path terrain values and the fact that a climbing/descending aircraft fails to match the assumed locations. This was corrected by efficiently interpolating the values during the trajectory generation process. The technique is as follows:

> In determining the lateral path, a nearest neighbor selection process is used to find the terrain altitude associated with the location of each point in the trajectory.
>
> Next, a linear interpolation is made at one foot intervals across the patch length used for vertical trajectory generation. Then during vertical trajectory computation, terrain values are simply found by using a table look-up.

This technique significantly increased the speed of the vertical trajectory process, while more accurately placing the nodes.

2.4.1.9 Dynapath Source Code Delivery

The modified decoupled Dynapath source code was delivered to NASA on June 6, 1986. The software is coded in Fortran 77. It is designed to operate on the VAX/VMS operating system and, in fact, was implemented without major modification.

There are seven major software modules to the delivered code. The sections are:

- Section 1: The driver routine
- Section 2: Major lateral optimization routines
- Section 3: Major vertical optimization routines
- Section 4: Common blocks used in the algorithm
- Section 5: Data files
- Section 6: Utilities to read data files
- Section 7: Compile and link command files

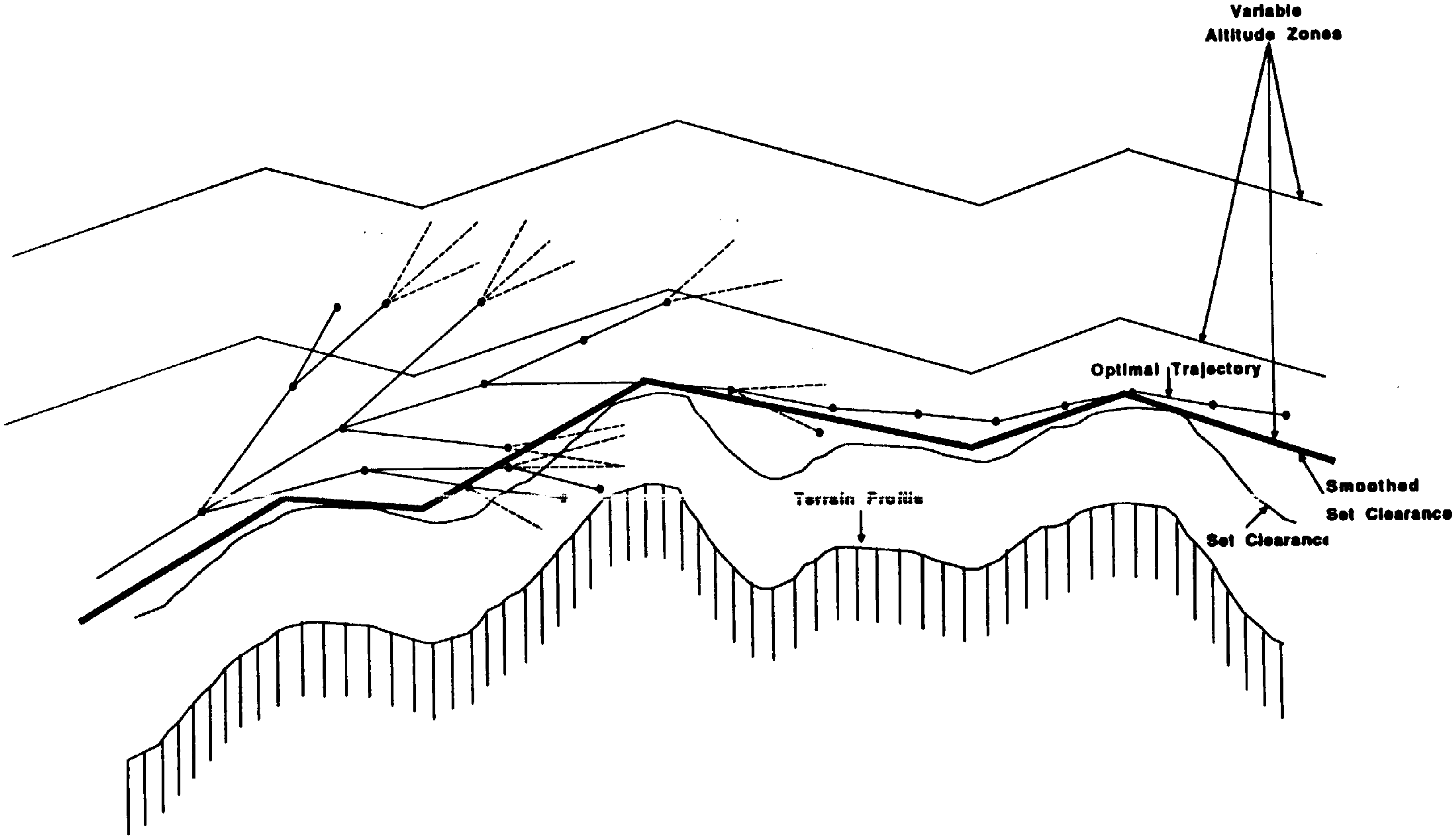


Figure 41. Refined Vertical Path Determination

The driver section of the program first executes the lateral optimization module, then the vertical optimization section. Sections 4 through 7 are program and data support components for the integrated software system. The Appendix details each routine and briefly describes the functions.

2.4.1.10 Continued Dynapath Development

The development of the Dynapath code has continued at an intensive level under NASA-SBIR Contract NAS2-12402. Developments under this Phase II contract have resulted in significant computational speed enhancements. Adaptations were made to the TF/TA ratio element of the cost module to command the trajectory smoothly through waypoints. Other developments were pursued to closely support a real-time man-in-the-loop simulation using Dynapath in October 1986.

Figure 42 illustrates the execution of Dynapath for a typical scenario used in the real-time simulation. The commanded lateral trajectory is superimposed on the NASA-NOE terrain data base for a simulated NOE helicopter flight. The polygons represent synthetic mountains. The degree to which the commanded trajectory deviates from the route segment centerline is determined by the TF/TA ratio and the height of the terrain. The adaptability of the trajectory to maneuver around waypoints and obstacles is limited by the performance constraints of the helicopter.

Dynapath continues in development under this contract with the principal goals being to support real-time NOE flight capability and to enhance the pilot-vehicle interface associated with using automated TF/TA techniques.

2.4.2 Far Field Navigation

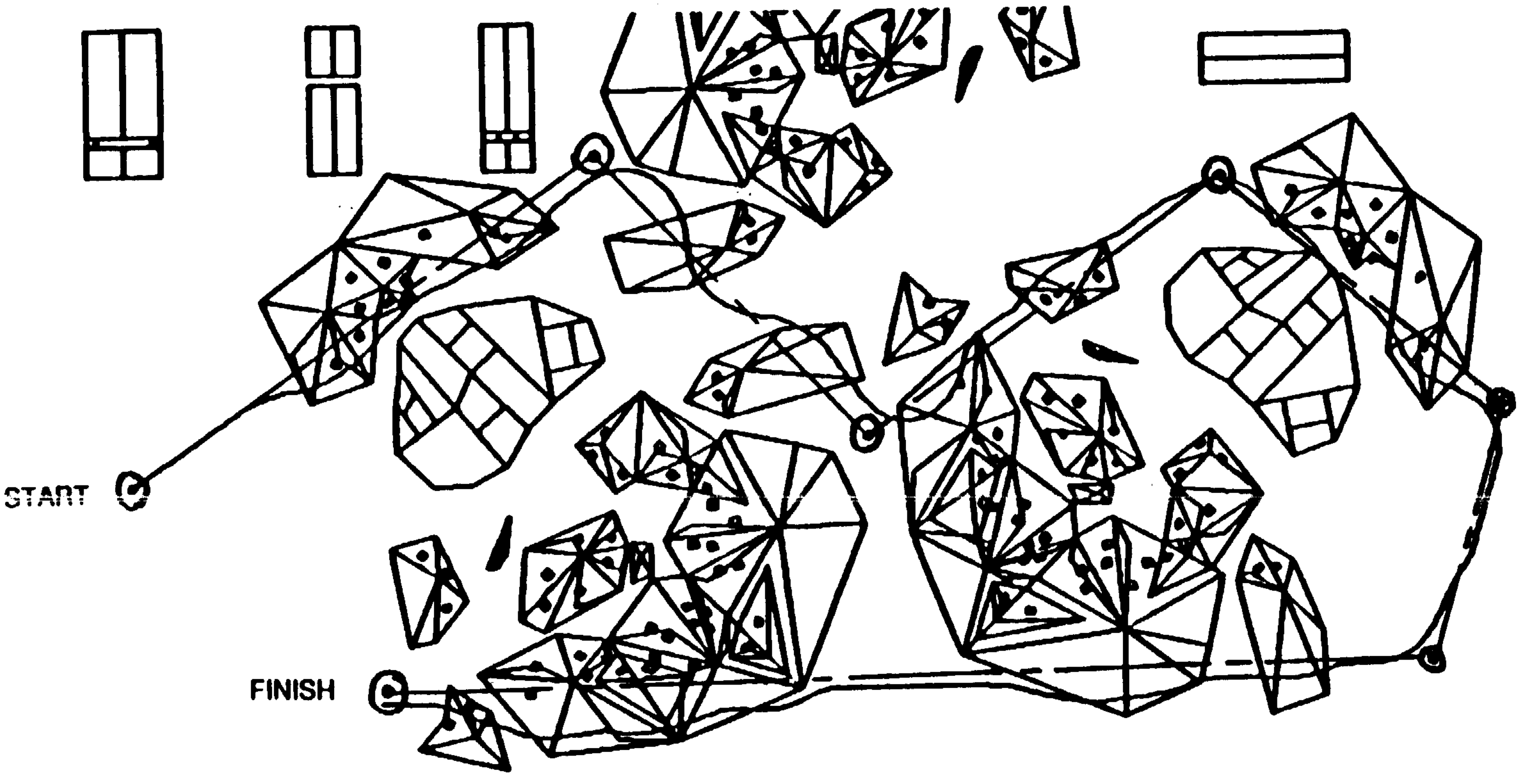
2.4.2.1 Dynaplan

The role for far field navigation was briefly discussed in a previous section of this report (Section 2.1.7). The purpose of far field navigation is to select the best "big picture" path to fly subject to the multiple constraints and goals of the mission.

A prototype mission planning capability has been developed to support this far field navigation element. The software, Dynaplan, is based on Dynamic Programming techniques which serve to optimize a cost functional. The process itself guarantees global optimality subject to the values (terrain, threats, etc.) placed on the cost components of the problem.

The method indicates promise both for use in an automated ground based mission planning function and also to serve as an on-board replanning capability. It serves as an ideal precursor to near field navigation by automatically generating a flyable coarse route which can be fine-tuned in real-time by Dynapath-type software.

Table 14 lists the key mission planning parameters. Terrain, threat, aircraft performance, and path constraints form the data base information which is managed and manipulated in generating the cost function for the optimization process. Mission specific items, iteration parameters, and evaluation criteria



NASA-NOE TERRAIN DATA BASE WAYPOINT SET 7

Figure 42. Lateral Path Generated for NASA-NOE Terrain Data Base Waypoint Set 7

reflect the human judgment and priorities with respect to the components of the problem and greatly affect the optimization process.

Figure 43 is a sample of the top-down display of a Dynaplan solution. Several optimized routes are shown to a target corresponding to different entry points to the target region. A commanded route is shown for comparison with this optimized route. The hazards and cost components of each route can be extracted for comparison/evaluation.

Table 14. Mission Planning Parameters

DATA BASE

DIGITAL ELEVATION DATA BASE
- DMA
- USGS

PATH CONSTRAINTS
- WEATHER
- THREATS
- WAYPOINTS
- INTERVISIBILITY

THREAT DATA
- LETHALITY
- INTERVISIBILITY
- DETECTION RADIUS
- ASPECT DEPENDENCY

PERFORMANCE DATA
- NOMINAL SPEED
- FUEL CONSUMPTION
- RANGE

EVALUATION

ROUTE PERFORMANCE
- TIME
- FUEL CONSUMED
- DETECTION PROBABILITY
- SURVIVAL PROBABILITY
- WEIGHTED COST

DISPLAY

OPTIMIZED ROUTE

RANGE/ALTITUDE/THREAT PROFILE

MISSION SPECIFIC

INITIAL CONDITIONS
- DESTINATION
- MAP LIMITS

WEIGHTING PARAMETERS
- MINIMUM TIME
- THREAT EXPOSURE
- DETECTION
- WEATHER

ITERATION PARAMETERS

MISSION GOALS

WAYPOINT REVISIONS

FUEL/TIME CONSTRAINTS

ALTERNATIVE ROUTES

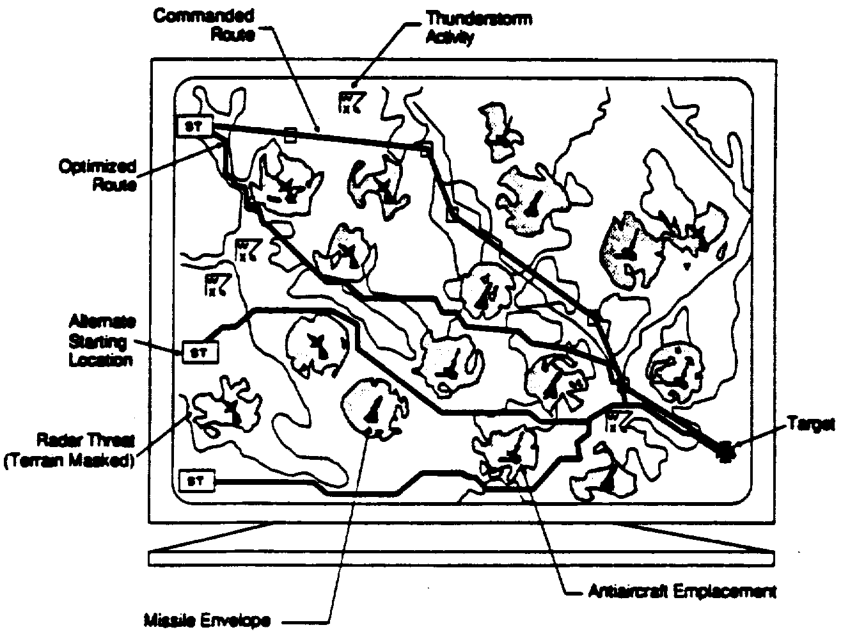


Figure 43. "Artistic" Output From Dynaplan

2.4.2.2 VAX Global Path Optimization Software Delivered to NASA

The prototype of the Dynaplan software was delivered to NASA for use in evaluating the applicative global algorithm to far field navigation. The software components supplied include the following:

- Cost generation module
- Optimization module (before modifications for maximum speed)
- Path retrieval module
- Vehicle model
- Simple (circular) threat module
- Input/output routines

These modules have been significantly enhanced in the generation of the Dynaplan Mission Planning Tool, but the keystone algorithms remain similar to the current product.

3.0 CONCLUSIONS

The efforts on this study, "Autonomous Flight and Remote Site Landing Guidance Research for Helicopters," have resulted in measurable developments in the technology required for supporting automatic helicopter flight in the NOE regime. There are three principal benefits which have come from this research:

1. Definition of a system concept which supports current and future work in this area.

2. Comprehensive research in the application of optical flow techniques to perform passive ranging on data within an imaging sensor field-of-view.

3. Development and delivery of automatic guidance algorithms and code which supports key concepts for NOE flight.

The system concept supports the desire by NASA to maximize the use of relatively low cost, passive sensors. By utilizing a digital map database as a sensor and integrating it with INS/GPS positional measurement, range measurements can be provided to the very near field navigation system with respect to the gross features of the landscape. The accuracy of this approach must be determined. Low power active systems and high resolution imaging sensors are necessary primarily to support measurement of hazards in the near field, particularly, the 10-1000 feet range.

Furthermore, the system concept supports driving active sensors with preliminary measurements from passive sensors. This approach is preferred to any generalized sensor blending/fusion techniques.

Active sensors may be a liability, making the rotorcraft visible to threats in a hostile military environment. The most desirable active ranging device, a CO_2 laser, can be hazardous to any nearby persons. However, because of the relatively low sensor range requirements (~1 km), the system should be significantly more difficult to detect than those designed for high speed/altitude operating environments.

Motion cueing of imaging sensors is the key to extracting the relative position and velocity of the rotorcraft as it moves across the terrain. By using image processing techniques to find key features in the terrain and collecting the same features over several sensor frames (or "snapshots") the optical flow, or 2-dimensional angular velocity of these features can be used to infer the relative position and velocity of the aircraft with respect to these features. The accuracy of the method is subject to the repeatability in collecting significant features and the ability to remove or account for sensor platform errors. It is strongly recommended that the sensor platform be stabilized in all 3 axes. Accurate acceleration measurements are needed to damp out positional errors.

The underlying technical base for individual sensor processing needs continuing research and development. The initial development of an optical flow vision approach applied to a single possible sensor system in this study shows promise, but also suggests the overall amount of study which is necessary to investigate the possibilities and flaws in this and related (e.g., stereo) machine vision concepts.

Additionally, it is important to stress the need to evaluate these sensor concepts using realistic data. The Ft. Rucker video and FLIR imagery used in this effort highlighted the problems of obtaining such data and converting it to an acceptable format.

In the realm of automatic guidance, real-time TF/TA software code, Dynapath, has been implemented and shows promise in simulations. Eventually, it is hoped that flight test will demonstrate relieving the pilot workload in low altitude TF/TA maneuvering. A wealth of practical details need to be addressed in such areas as pilot-vehicle interface, real-time computation loading in an aircraft system, in order to transition this technology into the practical aviation world.

Lastly, the far-field navigation technology is maturing into a practical tool to interface with the real-time near-field navigation. Concepts for such integration were illustrated in this effort.

Recommendations for Further Study

The Phase I effort of the autoguidance study has been successful in identifying promising concepts for using a relatively low cost passive sensor system and advanced real-time navigation techniques to enhance the safety of flight in the NOE environment. The key navigation algorithms are undergoing further refinement and maturity, under a different NASA contract.

There are several sensor-oriented areas of research which would greatly benefit from a Phase II follow-on effort. These items are outlined below:

Object Correlation. The passive ranging technique pursued in the study to date has been one of feature detection. Object correlation promises a more accurate ranging measure with little, if any, compromise in processing speed.

Data Display. Methods need to be determined as to how best present the predicted object avoidance information both to the pilot and autopilot system. Considerations of false alarms, noise, and conflicting information must be accounted for. There is also a need to clearly and simply direct information via the HUD or other means which do not require visual diversion from outside the cockpit.

Data Consistency. Greater attention should now be directed to applying the passive ranging techniques to sequences of data. By evaluating the consistency of information extracted form multiple images, and for a variety of scenarios, an understanding can be made as to the probabilities of detection/false alarm.

Navigation Data Estimation. In lieu of actual companion position and velocity information, off-line image processing techniques might be applied to enhance the estimates used for the velocity vector location. Either this refinement or actual data is essential for validating the results of passive ranging technique.

Sensor Requirements for Future Evaluation

The following recommendations should also be considered with regard to data collection for passive sensing:

- AGC Slow or Locked - especially on FLIR systems, have an extremely short time constant which creates unwanted contrast differences in successive images.

- Stable Video Sync - Many cameras or tape copying techniques severely degrade the sync, rendering the images unusable due to resulting litter.

- Standard Video Format (480 Lines) - Most FLIR systems tend to have approximately 780 video lines while commercial image processing systems are set up for 480 or 512 lines. Though the data can be converted, the equipment is not readily available.

- If Film Input, Then Synchronized Conversion to Video - Film is an excellent data source because of its sharpness and exposure latitude. However, to avoid image blurring, each video frame must contain only one film frame.

- No Symbology on Input Video - Many currently available data sources have symbology superimposed on the image. This symbology overlaps critical data or greatly increases the image processing requirements.

- Concurrent Navigation Data - Camera attitude and position information is required for complete validation of the results.

- Evaluation of Aircraft Flexing - If the inertial system is separate from the camera, then the flexibility between the two must be known to maintain sufficient attitude accuracy.

REFERENCES

1. Denton, R. V., J. E. Jones, R. P. Denaro, "Use of the DMA Digital Terrain Elevation Data Base for Flight Trajectory Generation, Terrain Following/Terrain Avoidance, and Weapon Delivery," AGARD Lecture Series No. 122, Application of Digital Mapping Technology to Guidance and Control Systems, March, 1983.

2. M. J. Wendl, J. E. Wall, and G. D. Young, "Advanced Automatic Terrain Following/Terrain Avoidance Control Concepts Study," 1982 NAECON Conference Proceedings.

3. Denton, R. V., and J. P. Marsh, "Application of Autopath Technology to Terrain/Obstacle Avoidance," NAECON Proceedings, 1982.

4. Denton, R. V., J. E. Jones, M. Bird, and M. D. Olinger, "The Trajectory Generator for Tactical Flight Management," NAECON Proceedings, 1983.

5. Y. Barniv, P. L. Cowell, R. V. Denton, R. L. Farmsworth, M. A. Grossberg, J. P. Marsh, D. R. Morgan, C. A. Reid, "Cruise Missile Path Optimization - Phase II," Final Technical Report, Systems Control, Inc., September 1980.

6. W. J. Murphy, G. D. Maroon, R. A. Kupferer, D. J. Halski, W. G. McDonough, and T. J. Mascheck, "Tactical Flight Management Exploratory Development Program - Interim Technical Report AFWAL-TR-3023," April 1984.

7. M. J. Wendl, D. R. Katt, B. B. Barnes, and S. Cooper, "Advanced Automatic Terrain Following/Terrain Avoidance Control Concepts, Volume I," CONFIDENTIAL, Mcdonnell Aircraft Company, St. Louis, MO, February 1984.

8. W. W. Harrington, T. G. Gates, D. E. Russ, "TF/TA System Design Evaluation Using Pilot-in-the-Loop Simulation: The Cockpit Design Challenge," SAE Aerospace Congress and Exposition, October 1984.

9. Sensor Blending for Terrain and Obstacle Avoidance, AFWAL Contract F33615-82-C-1863 with Boeing Military Airplane Company.

10. Denton, R. V., J. E. Jones, P. L. Froeberg, and R. Nikoukhah, "Terrain Following/Terrain Avoidance Algorithm Study," AFWAL/FIGL, Final Report, May 1985.

11. Aggarwal, J. K., "Motion and Time-Varying Imagery - An Overview," Proceedings of Workshop on Motion: Analysis and Representation, May 7-9, 1986, Charleston, SC.

12. Kearney, J. K., W. B. Thompson, and D. L. Boley, "Optical Flow Estimation: An Error Analysis of Gradient-Based Methods with Local Optimization," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 2, March 1987.

13. Jain, R., S. L. Bartlett, and N. O'Brien, "Motion Stereo Using Ego-Motion Complex Logarithmic Mapping," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 3, May 1987.

14. R. V. Denton and L. McGee, "Automation of Guidance for NOE Flight," Autonomous All-Weather Navigation and Landing Workshop, Naval Postgraduate School, January 1986.

15. R. V. Denton, N. Pekelsma, M. Hagen, L. McGee, "Guidance Automation for Nap-of-The-Earth Flight," 7th Digital Avionics Systems Conference, October 1986.

16. D. Bailey, N. Pekelsma, "A Pilot's Perspective of Automated Mission Management," 43rd American Helicopter Society Forum, to be published.

17. F. W. Smith and M. Streicker, "Passive Ranging from a Moving Vehicle via Optical Flow Measurement," August 1987, SPIE.

APPENDIX

DYNAPATH SOURCE CODE

Section 1: MAIN.FOR — Main program both for lateral and vertical trajectory generation

Section 2: DPH.FOR — The driver subroutine which performs the optimization necessary to find the two dimensional solution. The following procedures are invoked by DPH.FOR.

- CONTROLS.FOR — Computes current lateral control set
- ICS.FOR — Computer initial condition vectors
- PRUNEH.FOR — Tree pruning procedure
- TREEH.FOR — Generates n-level trees in X-Y axis where 0 < n < 8
- RECOVER.FOR — Retraces optimal solution from winner node to creator through all trees
- LEAFH.FOR — Generates offspring nodes for the trees and tests hard constraints
- LIMITS.FOR — Examines validation of input parameters
- LINE.FOR — Draws a line from point-1 to point-2
- DRAWLINE.FOR — Graphically displays the region in which the Dynapath algorithm is run

Section 3: VDPH.FOR — The driver subroutine which performs the optimization necessary to find the vertical trajectory. The following procedures are invoked by VDPH.FOR

- VCNTRLS.FOR — Computes current vertical CONTROL.SET
- VLEAFH.FOR — Vertical LEAFH generator
- VTREEH.FOR — Vertical tree generator
- VPRUNEH.FOR — Vertical tree pruning procedure
- VRECOVER.FOR — Retraces vertical optimal solution from winner node to creator node through all trees
- INIT_ARRAY.FOR — Initialize vertical Dynapath arrays
- LINEAR.FOR — Performs linear interpolation
- NEWSETCL.FOR — Generates a new set clearance

VINDEX.FOR	Determines distance, altitude, and angular indices
SINETBL.FOR	Fills an array of sin(angle) where -76 < angle < 76
COSITBL.FOR	Fills an array of cos(angle) where -76 < angle < 76

Section 4: COMMON.BLOCKS

MAINPROG.FOR	Dynapath definitions
SCEN.FOR	Common blocks for scenario variables
LATERAL.FOR	Common blocks for lateral Dynapath
VERTICAL.FOR	Common blocks for vertical Dynapath

Section 5: INPUT DATA FILES

ICIN.DAT	Dynapath scenario files
SINE.DAT	Sine table
COSINE.DAT	Cosine table
DYN.DAT	512 by 512 data file (528 blocks). Each record contains 8192bytes. The first record is the header record.

Section 6: OUTPUT DATA FILES

DPH.DAT	Solution vector both for lateral and vertical Dynapath
PATCH.DAT	Patch update solution vector
PERF.DAT	Performance statistics in vertical trajectory
SET.DAT	Interpolated terrain altitude values for vertical trajectory

Section 7: UTILITY FOR READING DATA

READALTI.FOR	Reads an array of 512 by 512 of altitude values which are declared to be logical *1

Section 8: COMPILATION/LINKAGE

COMPILE.COM	Command file
LINK.COM	Command file

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.
NASA CR-177478

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
AUTONOMOUS FLIGHT AND REMOTE SITE LANDING GUIDANCE RESEARCH FOR HELICOPTERS

5. Report Date
AUGUST 1987

6. Performing Organization Code

7. Author(s)
R.V. DENTON, N.J. PECKLESMA, AND F.W. SMITH

8. Performing Organization Report No.

10. Work Unit No.

9. Performing Organization Name and Address
TAU CORP.
485 ALBERTO WAY, BUILDING D
LOS GATOS, CA 95030

11. Contract or Grant No.
NAS2-12180

13. Type of Report and Period Covered
CONTRACTOR REPORT

12. Sponsoring Agency Name and Address
NASA AMES RESEARCH CENTER
M.S. 210-9
MOFFETT FIELD, CA 94035

14. Sponsoring Agency Code
505-66-11

15. Supplementary Notes
POINT OF CONTACT: LEONARD A. MCGEE
M.S. 210-9
NASA AMES RESEARCH CENTER
MOFFETT FIELD, CA 94035
(415) 694-5443 or FTS 464-5443

16. Abstract
The objective of this study was to conduct research that has the potential of leading to automated low-altitude flight and landing in remote areas within a civilian environment, where initial cost, on-going maintenance costs, and system productivity are important considerations. An approach has been implemented which has: 1) utilized those technologies developed for military applications which are directly transferable to a civilian mission; 2) exploited and developed technology areas where new methods or concepts are required; and 3) undertaken research identified as having the potential of leading to innovative methods or concepts required to achieve a manual and fully automatic remote area low-altitude and landing capability. The project has resulted in a definition of a system operational concept that includes a sensor subsystem, a sensor fusion/feature extraction capability, and a guidance and control law concept. These subsystem concepts have been developed to sufficient depth to enable further exploration within the NASA simulation environment, and to support programs leading to flight test.

17. Key Words (Suggested by Author(s))
AUTOMATED LOW-ALTITUDE FLIGHT,
AUTOGUIDANCE TECHNOLOGY, SENSOR FUSION,
OBSTACLE DETECTION, OBSTACLE AVOIDANCE

18. Distribution Statement
UNLIMITED DISTRIBUTION

STAR CATEGORY 08

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
UNCLASSIFIED	UNCLASSIFIED	97	